



StreamNow

*Simulación de una red de streaming
educativo con ns3*

Ángel Donet Granado
Miguel García Guzmán
Cristóbal Romero Lama
v2.0
Grupo 04

Análisis de ancho de banda de StreamNow

Indice:

1. Introducción	3
2. Análisis del Sistema Inicial	4
2.1 Arquitectura de la Red	5
2.1.1 Equipos	5
2.1.2 Protocolos	5
2.1.3 Especificaciones Técnicas: Equipos y enlaces	6
2.1.4 Tráfico de Usuarios	7
2.1.5 Técnicas QoS usadas	8
Marcado de Prioridades (DSCP)	8
Control de Tráfico y Ancho de Banda	9
2.2 Servicios Proporcionados en el Sistema	9
3. Planteamiento del problema	11
4. Objetivos	11
5. Planteamiento del Proceso de Trabajo	12
5.1 Organización del Proceso de Simulación	12
5.2 Recursos Necesarios	13
6. Especificación del Modelo	14
6.1 Modelo de la Topología y Simplificaciones	14
6.1.1 Nodos	14
6.1.2 Enlaces y conexiones con el ISP	16
6.1.3 Asignación de direcciones IP	17
6.2 Modelo de Tráfico y Simplificaciones	17
6.2.1 Tráfico de video y QoS	17
6.2.2 Tráfico de usuarios	17
9. Diseño preliminar de los experimentos	19
9.1. Parámetros Invariables	20
9.2. Parámetros Configurables y de Control	20
9.3. Factores de Estudio (Variables Dinámicas)	20
9.4. Medidas de Interés (Variables Dependientes)	20
10. Preparación de los datos de entrada	21
10.1. Escenario de Calibración	21
10.2. Proceso Iterativo de Calibración de Retardos	21
10.3. Valores de Retardo Finales	21
11. Implementación del modelo	22
11.1. Identificación y creación de los elementos de red	23
11.2. Configuración del Tráfico y las Aplicaciones	25
11.3 Configuración y Control de la Simulación	26

11.4. Medición de Métricas de Interés	27
11.5. Arquitectura del programa de simulación	27
12. Verificación y validación	33
12.1. Verificación	33
12.1.1. Verificación de la Topología y el Enrutamiento	33
12.1.2. Verificación del Modelo de Tráfico	34
12.1.3. Verificación del Flujo Experimental y la Recolección de Datos	35
12.2. Validación	35
12.2.1. Objetivo y Metodología de Validación	35
12.2.2. Resultados del Experimento de Validación	36
12.2.3. Interpretación y Conclusiones de la Validación	36
13. Diseño final de los experimentos	37
13.1. Consideraciones Generales de la Simulación	37
13.2. Gráficas Esperadas	37
13.2.1. Gráfica Principal: Bitrate Mínimo Requerido vs. Número de Usuarios	37
13.2.2. Gráficas de Métricas de QoS (Retardo, Jitter, Pérdida de Paquetes) vs. Bitrate	38
14. Experimentación	38
15. Análisis e interpretación de los resultados	46
16. Conclusiones	49
17. Referencias	50

Índice de figuras:

<u>Figura 1</u> (Gráfica que representa el aumento de los usuarios en los últimos años)	4
<u>Figura 2</u> (Escenario real de StreamNow)	5
<u>Figura 3</u> (Representación gráfica del patrón On/off)	9
<u>Figura 4</u> (Diagrama de paso de mensajes de solicitudes de vídeo por parte del usuario)	17
<u>Figura 5</u> (Representa el escenario ya modelado, sobre el que vamos a trabajar)	21
<u>Figura 6</u> (Representa los enlaces y sus bitrates en el modelo)	23
<u>Figura 6-2</u> (Gráfica final bitrate/usuarios)	24
<u>Figura 7</u> (Representa bitrate y delay de los enlaces del modelo)	27
<u>Figura 8</u> (Elementos ns3 implicados en el modelo)	28

<u>Figura 9</u> (Bitrate y delay de enlaces del modelo)	
<u>Figura 10</u> (patrón ON OFF)	29
<u>Figura 11</u> (Diagrama de flujo)	33
<u>Figura 12</u> (Clases implicadas en el código)	35
<u>Figura 13</u> (Valores finales de retardo en los enlaces del modelo)	36
<u>Figura 14</u> (Gráfica porcentaje de pérdida de paquetes por cada conjunto de usuarios//bitrate)	37
<u>Figura 15</u> (Gráfica retardo por cada conjunto de usuario//bitrate)	38
<u>Figura 16</u> (Gráfica del jitter por cada conjunto de usuario//bitrate)	38
<u>Figura 17</u> (Gráfica final bitrate/usuarios)	39
<u>Figura 18</u> (Margen error retardo)	44
<u>Figura 19</u> (Margen error jitter)	45
<u>Figura 20</u> (Margen error pérdida de paquetes)	
(Figura 20) Gráfica porcentaje de pérdida de paquetes por cada conjunto de usuarios//bitrate	43
<u>(Figura 21)</u> Margen error de retardo	44
<u>(Figura 22)</u> Margen error de jitter	45
<u>Figura 23</u> (Margen error pérdida de paquetes)	46

Índice de tablas:

<u>Tabla 1</u> (Clasificación del tráfico según su bitrate)	11
<u>Tabla 2</u> (Marcación DSCP según la clase de tráfico)	11
<u>Tabla 3</u> (Tasa máxima según la clase)	13
<u>Tabla 4</u> (Asignación de CIR y PIR según calidad)	14
<u>Tabla 5</u> (Probabilidad de descarte según calidad)	15

<u>Tabla 6</u> (Condiciones de calidad mínimas exigidas por la empresa)	12
<u>Tabla 7</u> (Tabla de medias de pérdida de paquetes, retardo y jitter por cada conjunto de usuarios//bitrate)	36
<u>Tabla 8</u> (Condiciones de calidad mínimas exigidas por la empresa)	47
<u>Tabla 9</u> (Resultados característicos de la gráfica final)	48

1. Introducción

Este documento presenta un estudio de simulación para la empresa de streaming educativo StreamNow S.A. El objetivo es ayudarles a planificar el crecimiento de su red.

El contenido principal es un análisis de red realizado con el simulador ns-3. Se describe el modelo creado para representar el sistema de StreamNow, los experimentos realizados variando el número de usuarios y el ancho de banda, y los resultados obtenidos en cuanto a la calidad del servicio.

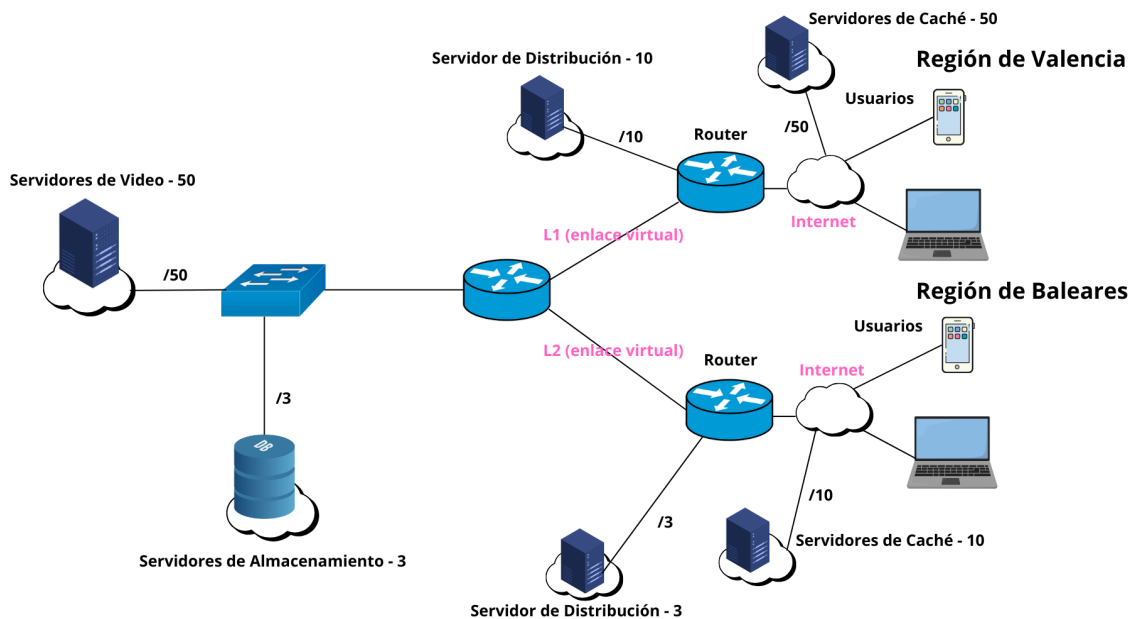
El objetivo final es entregar a StreamNow una estimación precisa del ancho de banda mínimo que necesitan contratar para su enlace de Valencia según el número de usuarios (de 100 a 500), asegurando que se cumplen unos requisitos mínimos de calidad establecidos por la propia empresa.



(Figura 1)- Gráfica que representa el aumento de los usuarios en los últimos años

2. Análisis del Sistema Inicial

La infraestructura de StreamNow se divide en un centro de datos principal y dos regiones geográficas a las que da servicio: Valencia y las Islas Baleares. La siguiente figura muestra la arquitectura general de la red:



(Figura 2)-Escenario real de StreamNow

Observaciones importantes

- Los guiones seguidos de un número (ej: -3) representan el número de dispositivos, y las barras seguidas de un número (ej: /10) representan el número de enlaces en esa conexión.
- Los enlaces L1 y L2 son enlaces virtuales contratados a un proveedor externo (ISP) para conectar la red con los usuarios. Nuestro estudio se centrará en el enlace L1 de Valencia.

2.1 Arquitectura de la Red

2.1.1 Equipos

La red de StreamNow SA está compuesta por varios elementos:

- **Servidores de Almacenamiento:** Guardan todos los vídeos y contenidos multimedia de la plataforma.
- **Servidores de Vídeo:** Gestionan las peticiones de los usuarios y saben en qué servidor de almacenamiento está cada vídeo.
- **Servidores de Caché:** Almacenan temporalmente los vídeos más populares en cada región geográfica para que la entrega sea más rápida.
- **Servidores de Distribución:** Actúan como intermediarios (proxies) para distribuir el contenido que no está en la caché a cada región.
- **Routers:** Conectan las diferentes partes de la red y encaminan el tráfico.
- **Switch:** Conecta los servidores de almacenamiento y de vídeo.

2.1.2 Protocolos

La comunicación en la red de StreamNow se basa en un conjunto de protocolos estándar. Los más importantes son:

- **Nivel de Enlace:** En los tramos cableados se utiliza Ethernet, el estándar más común para redes locales (LAN). En concreto se usa la versión Ethernet IEEE 802.3bj [\[17\]](#)
- **Nivel de Red y Routing:**
 - **IPv4:** Se usa para dar una dirección única a cada equipo de la red y poder identificarlo. [\[18\]](#)
 - **RIP (Routing Information Protocol):** Es el protocolo de enrutamiento que usan los routers para descubrir las mejores rutas y poder enviar los datos desde el servidor hasta los usuarios de manera dinámica. [\[20\]](#)
- **Nivel de Transporte:** Se utiliza TCP (Transmission Control Protocol) para asegurar que los datos del vídeo se entregan de forma fiable, completa y en orden. [\[19\]](#)
- **Nivel de Aplicación:**
 - **HTTPS:** El servicio de vídeo se transmite sobre HTTPS para que la comunicación sea segura y cifrada. [\[32\]](#)
 - **DASH:** Se usa esta tecnología para el streaming adaptativo, que permite ajustar la calidad del vídeo en tiempo real según el ancho de banda del usuario, mejorando su experiencia. [\[21\]](#)

2.1.3 Especificaciones Técnicas: Equipos y enlaces

El hardware de StreamNow está diseñado para soportar un alto rendimiento, necesario para el streaming de vídeo.

Equipos

- **Servidores (Dell PowerEdge R740xd):** Son equipos muy potentes, con procesadores multinúcleo y almacenamiento rápido (discos SSD). Se encargan de procesar y servir los vídeos a los usuarios. [13]
- **Routers (Cisco Catalyst 9500):** Son routers de gama alta, con puertos de gran capacidad (10 Gbps) para gestionar el enorme volumen de tráfico que genera el streaming de vídeo. [14]
- **Switch (Cisco Catalyst 9500 Series):** Al igual que los routers, este switch ofrece puertos de muy alta velocidad (10 Gbps y 40 Gbps) para interconectar de forma eficiente todos los servidores en el centro de datos. [15]

Enlaces

- **Enlaces de la red interna:** Las conexiones que unen los servidores con el switch y los routers, utilizan enlaces de fibra óptica 10GBASE-LX4 de hasta 10 Gigabits por segundo (10GbE). Esto garantiza que no haya cuellos de botella dentro de la propia infraestructura de StreamNow. [16]
- **Enlace virtual L1:** Tiene una capacidad contratada de 10 Mbps. Es el enlace de interés y su bitrate será modificado durante nuestro estudio.
- **Enlace virtual L2:** Tiene una capacidad contratada de 1Gbps.

2.1.4 Tráfico de Usuarios

Como características fundamentales del tráfico del sistema, podemos decir que no todos los usuarios requieren el mismo bitrate. StreamNow ofrece 3 calidades de vídeo distintas, FullHD, HD y SD, sus porcentajes son los siguientes:

- **FullHD:** 70%
- **HD:** 20%
- **SD:** 10%

También podemos caracterizar el tráfico respecto al de subida y de bajada, al ser un servicio de video la mayor parte va a ser de bajada, en el caso de StreamNow tenemos:

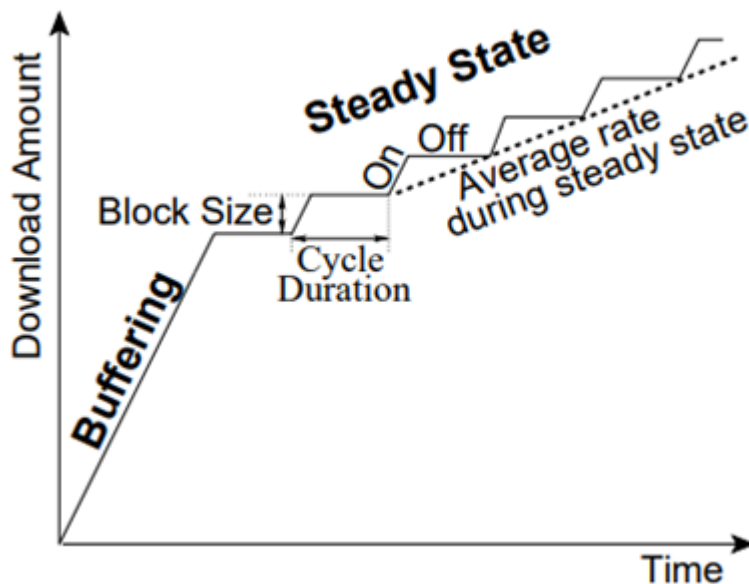
- **Bajada:** 97,7%
- **Subida** 2,3%

Estos parámetros han sido proporcionados por la empresa.

Como es normal en las transmisiones de video, los usuarios no requieren el mismo bitrate de datos durante la transmisión de video, el buffer permite descargar parte del video antes de su visualización, normalmente suele ser en torno a unos 40s de video que se pre descarga para su posterior visualización. Esta característica hace que el flujo de datos entre cliente y servidor siga un patrón ON/OFF. Se puede ver en [1]

El patron ON/OFF consiste en lo siguiente:

1. **Buffering inicial:** Al comenzar la reproducción de un video, se descarga una porción del video en el buffer, suficiente para unos **40 segundos** de reproducción. Esta fase garantiza que el video pueda reproducirse sin interrupciones.
2. **Ciclos ON-OFF:** Una vez que el buffer inicial está lleno, el sistema alterna entre fases de descarga rápida (ON) y de inactividad (OFF), donde no se descargan datos pero el video sigue reproduciéndose desde el buffer. Durante la fase ON, se descarga más rápido que la tasa de reproducción del vídeo, asegurando que el buffer siempre tenga suficiente contenido.



(Figura 3)- Representación gráfica del patrón On/off.

Para poder modelar el tráfico del sistema hemos pedido a la compañía datos sobre la hora del día donde más tráfico de media hay y nos han informado que es desde

las 10:00 hasta las 11:00 de la mañana, donde cada usuario inicia sesión una media de **0,6 veces por hora** y permanece en el sistema una media aproximada de **5 minutos**. Además la latencia media es **5 ms**. Más adelante en el modelado se detallarán todos los datos recogidos con mayor precisión y su uso.

2.1.5 Técnicas QoS usadas

La estrategia de QoS aplicada en los routers de StreamNow se basa en dos conceptos fundamentales para gestionar el tráfico de vídeo: el marcado de prioridades y el control de las tasas de tráfico a través de políticas.

Marcado de Prioridades (DSCP)

En primer lugar se clasifica y marca cada paquete de vídeo según su importancia. A cada calidad de vídeo se le asigna un código de prioridad (DSCP) que los routers pueden interpretar. A mayor calidad, mayor prioridad.

Clase de Tráfico	Código DSCP	Nivel de Prioridad
Full HD (1080p)	AF41	Alta
HD (720p)	AF31	Media
SD (480p)	AF21	Baja

(Tabla 1)-Marcación DSCP según la clase de tráfico
 Los datos obtenidos del campo DSCP se encuentran en [26].
 Los datos de la tabla obtenidos se encuentran en [9].

Control de Tráfico y Ancho de Banda

Una vez que el tráfico está marcado, los routers aplican políticas para controlar su flujo. La más importante es la limitación de la tasa de transferencia para cada clase de servicio evitando que una sola clase de tráfico pueda saturar la red.

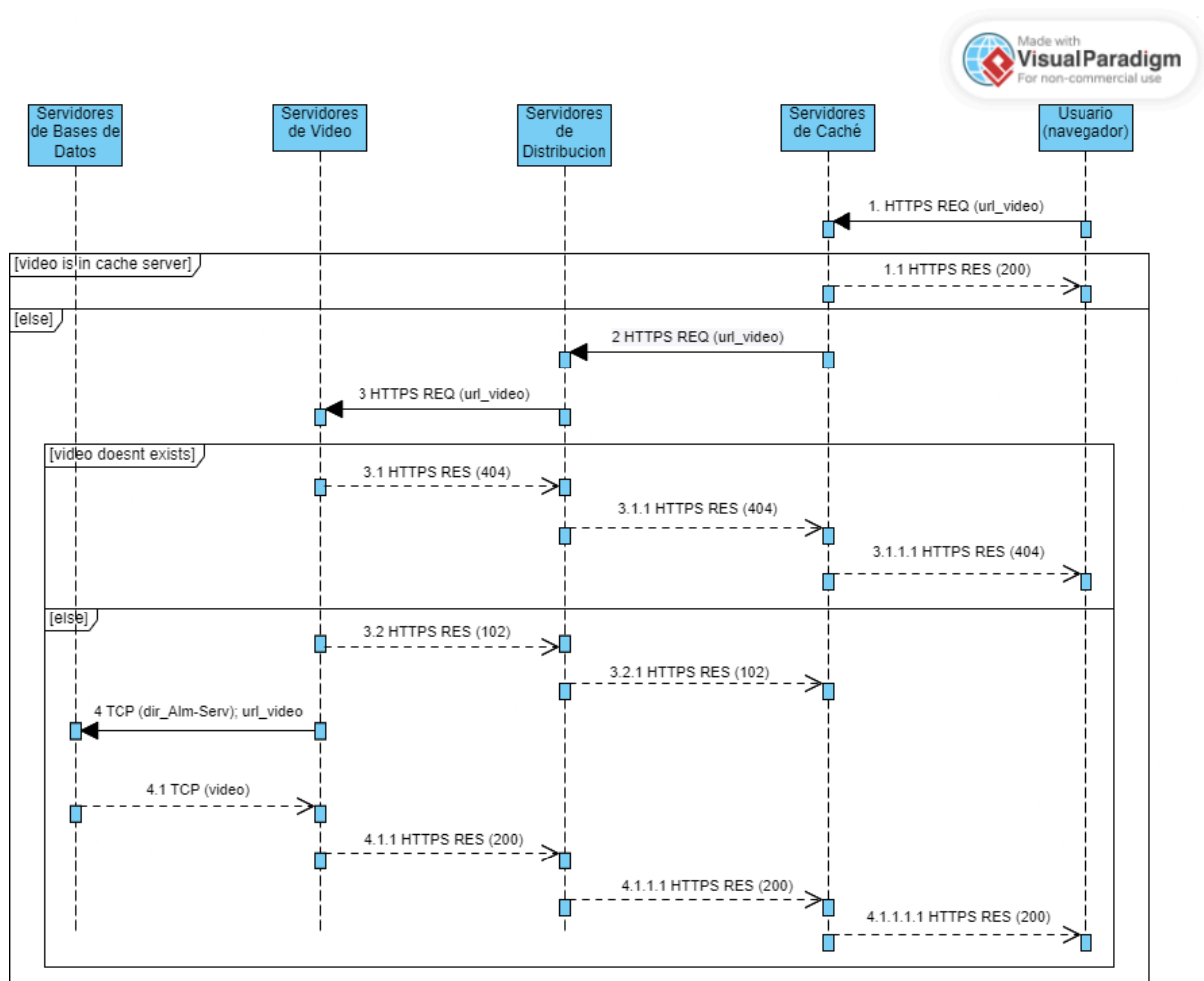
En concreto el sistema real tiene los siguientes límites:

Clase de Tráfico	Calidad de Video	Bitrate estimado
Clase 1	Full HD (1080p)	10 Mbps

(Tabla 2)- Clasificación del tráfico según su bitrate
 Datos de la tabla obtenidos de [8].

2.2 Servicios Proporcionados en el Sistema

Diagrama de secuencia



(Figura 4)- Diagrama de paso de mensajes de solicitudes de vídeo por parte del usuario.

Explicación detallada del servicio

Cuando un usuario solicita un video, la solicitud primero se dirige a los servidores de caché. Si el video está disponible en el servidor de caché, este lo entrega directamente al usuario. Si no lo está, el servidor de caché reenvía la solicitud a los servidores de distribución.

Los servidores de distribución, a su vez, envían una consulta HTTPS al servidor de video para verificar si el video existe. Si no se encuentra el video, el servidor de video responde con un mensaje HTTP 404 (Not Found), que es enviado de vuelta al servidor de distribución, luego al servidor de caché y, finalmente, al usuario. Si el video sí está disponible, el servidor de video envía un mensaje HTTP 102 (Processing) al servidor de distribución, que a su vez lo retransmite al servidor de caché. Esto pone a ambos en espera del contenido.

A continuación, el servidor de vídeo establece una conexión TCP con el servidor de almacenamiento que contiene el video solicitado. Una vez establecida esta conexión, el video se transfiere al servidor de video, desde donde es enviado al servidor de distribución, que a su vez lo reenvía al servidor de caché. El servidor de caché entrega el video al usuario y guarda una copia para futuras solicitudes. La distribución final del video utiliza los protocolos HTTPS y DASH para garantizar una transmisión eficiente y segura.

Los datos obtenidos del protocolo HTTP y DASH se encuentran en [32] y [21] respectivamente

Todos los enlaces de las referencias se encuentran al final del documento, donde [32] y [21] son el número de referencia

La red está organizada de forma que:

- El servidor de video mantiene una tabla que vincula las direcciones de los servidores de almacenamiento con las URLs de los videos.
- Los servidores de caché conocen las direcciones de todos los servidores de distribución lo cual le permite comunicarse con todos ellos.
- Los servidores de distribución conocen las direcciones de los servidores de video.

Esta estructura asegura una distribución eficiente del contenido, al mismo tiempo que mantiene un alto nivel de seguridad en la red.

3. Planteamiento del problema

StreamNow actualmente cuenta con aproximadamente 100 suscriptores y se estima alcanzar en un futuro en torno a los 500, lo que puede comprometer la estabilidad del servicio que ofrecen.

En concreto tienen mayor preocupación por la parte de la red que da servicio a Valencia ya que es donde más demanda reciben (estiman adquirir unos 150 usuarios en Baleares, mientras que en Valencia estiman unos 350).

Por todo esto, StreamNow desea estimar que ancho de banda contratar a la operadora que da salida a Valencia para un rango de crecimiento de 100 a 500 usuarios totales del sistema, asegurando unas mínimas calidades de servicio.

4. Objetivos

El objetivo será entregar una gráfica a StreamNow donde se represente el ancho de banda necesario a contratar a la operadora que da salida a Valencia para atender a un rango de 100-500 usuarios, asegurando unos requisitos mínimos de calidad de servicio especificados a continuación:

Parámetro	Valor máximo
Latencia (retardo)	20 ms
Jitter	8 ms
Pérdida de paquetes	0.5 %

(Tabla 3) - Condiciones de calidad mínimas

Para la obtención de estos datos de la tabla hemos buscado en diversas fuentes y las hemos comparado, eligiendo unos valores medios para obtener resultados realistas. Referencias [3], [4], [5], [6], [7].

Para escoger los valores de los requisitos, nos hemos basándonos en estándares de la industria consultando en varias fuentes los valores más típicos que equilibran la calidad de experiencia con la viabilidad técnica.

5. Planteamiento del Proceso de Trabajo

Para el estudio del proyecto utilizaremos simulaciones en NS3, una herramienta ampliamente empleada para modelar redes debido a su precisión y flexibilidad. La simulación nos permite analizar el comportamiento de un sistema complejo, como la red de StreamNow, sin necesidad de desplegar infraestructura física. Esto resulta más eficiente en términos de tiempo y recursos, además de permitir la evaluación de diferentes escenarios de carga y configuración de manera controlada.

5.1 Organización del Proceso de Simulación

El proceso de simulación se estructurará en varias etapas para garantizar resultados claros y coherentes. Cada etapa se enfoca en modelar, validar y analizar la red de

forma eficiente, permitiendo estudiar el ancho de banda necesario para satisfacer los requisitos de calidad del servicio en función del número de usuarios:

Etapla 1: Modelado del Sistema

Se creará un modelo simplificado del sistema real con el fin de, sin comprometer la validez de los resultados, centrarse exclusivamente en los aspectos más relevantes del estudio, como el comportamiento de la red bajo diferentes niveles de carga y los requisitos de calidad de servicio.

Etapla 2: Validación del Modelo

Se implementará el modelo en NS3 y se realizarán experimentos preliminares para verificar que el comportamiento simulado sea consistente con el sistema real.

Etapla 3: Experimentos Finales y Simulación

Se llevarán a cabo simulaciones en las que, para un número creciente de usuarios, se ajustará el ancho de banda hasta que se cumplan los requisitos mínimos de calidad de servicio. Este proceso se repetirá para diferentes escenarios de carga.

Etapla 4: Análisis de Datos

Los resultados obtenidos (ancho de banda necesario según el número de usuarios) se analizarán y se representarán mediante gráficos que permitan identificar las configuraciones óptimas para la red.

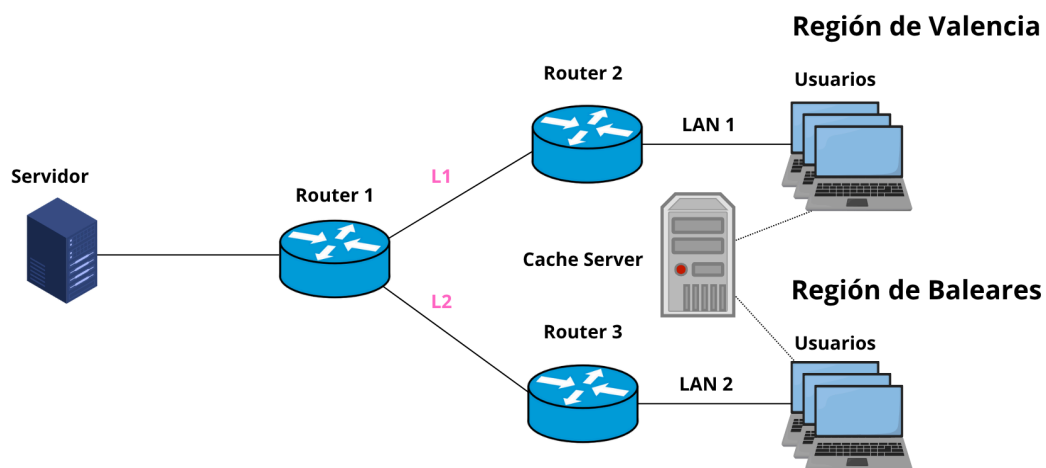
5.2 Recursos Necesarios

Para llevar a cabo este proyecto, contamos con los siguientes recursos:

- **Equipo humano:** Cuatro ingenieros de telecomunicación, encargados de diseñar, implementar y analizar los resultados de la simulación.
- **Equipamiento:** Cuatro portátiles de alto rendimiento que se utilizarán para ejecutar el simulador de redes y realizar los análisis.
- **Tiempo disponible:** El proyecto se desarrollará en un período de dos meses.
- **Herramientas:** El simulador de redes ns-3, una herramienta de código abierto que nos permitirá modelar la infraestructura de red de StreamNow y calcular el ancho de banda.

6. Especificación del Modelo

El modelo que se usará como referencia en todo el proceso de simulación es el siguiente:



(Figura 5). Representa el escenario ya modelado, sobre el que vamos a trabajar.

6.1 Modelo de la Topología y Simplificaciones

Para abordar de manera eficiente la simulación, se ha modelado la red de StreamNow teniendo en cuenta las siguientes simplificaciones sobre la topología:

6.1.1 Nodos

Servidor

- Cumple la función de los servidores de video y de almacenamiento
- Contendrá una única interfaz de red con 1Gbps de velocidad

Routers

- Uso del protocolo RIP para encaminamiento
- Una interfaz de red por salida con 1Gbps de velocidad.

Servidor de caché

- Todos los servidores caché se engloban en un único servidor de caché
- Contendrá una única interfaz de red con 1Gbps de velocidad

Usuarios

- El número de nodos de usuarios variará en cada simulación según las necesidades de estudio, irá desde 100 hasta 500 usuarios

Simplificación de los servidores de caché

Para simplificar el modelo, se ha decidido modelar los servidores de caché simplemente descartando un 60% del tráfico asumiendo que se consume en la propia LAN sin afectar al enlace. Esto significa que el servidor caché no será un objeto de ns3, aunque se ha puesto en la figura a modo representativo.

Para determinar el porcentaje descartado hemos consultado varias referencias y hemos contrastado sus valores. [\[36\]](#), [\[37\]](#)

Eliminación de servidores de distribución

Para simplificar el modelo, también se ha decidido eliminar los servidores de distribución del escenario. Dado que su función es parecida a proxies, su supresión no alteraría en gran medida los resultados que se obtengan en la simulación, y a cambio se reducirá la complejidad computacional necesaria para estudiar la red.

Eliminación del switch

Dado que los servidores de almacenamiento y video se han integrado en un solo nodo, y el switch tiene un rol secundario en este escenario, se ha optado por eliminarlo para simplificar el modelo.

Simplificación de las redes de usuarios: LANs

Las redes que conectan al ISP con los usuarios se han simplificado con dos enlaces que emularán redes LAN. Los usuarios de una misma región tendrán un rango de IPs asignados, simplificando así la gestión de direcciones.

Más adelante se mostrarán los distintos rangos de direcciones usados en todas las subredes del escenario.

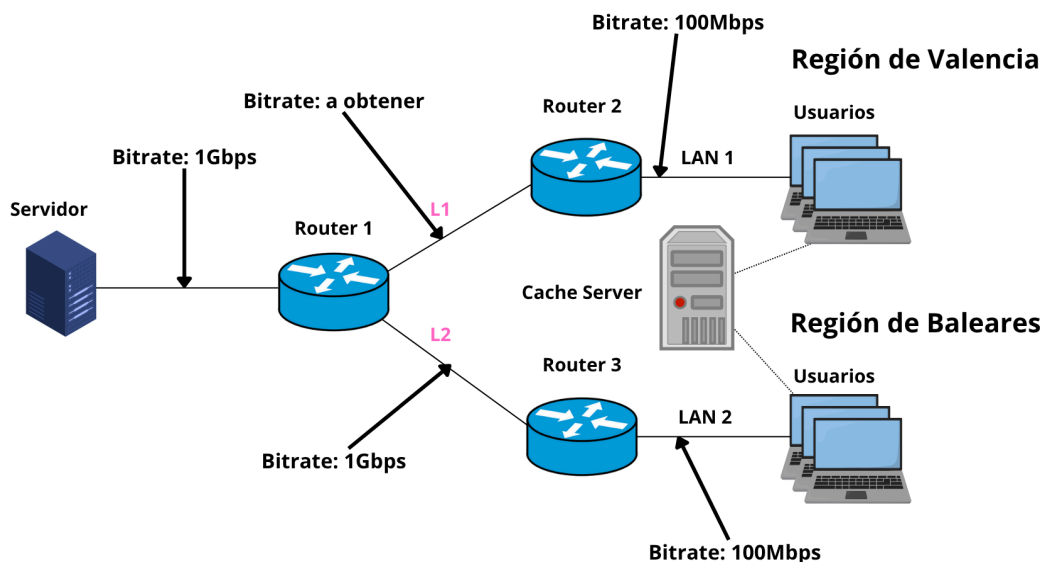
Bajada de bitrate de tráfico de video

Hemos bajado el bitrate de tráfico debido a que era inviable simularlo, ya que los tiempos de simulación eran excesivamente altos y no contamos con equipos específicos para simular. Ahora FullHD son 800kbps, HD 500 kbps y SD 200kbps.

Nota: Si se desea obtener unos resultados más realistas con un hardware especializado, en el código onoffRouting.cc, en `h.SetConstantRate(DataRate("....."))` añada los valores correspondientes indicados en la tabla 3.

6.1.2 Enlaces y conexiones con el ISP

Los enlaces de la red serán modelados con Ethernet IEEE 802.3bj. Además tendrán un retardo de transmisión definido (más adelante se detallan los valores), y una velocidad tal y como se muestra en la figura:



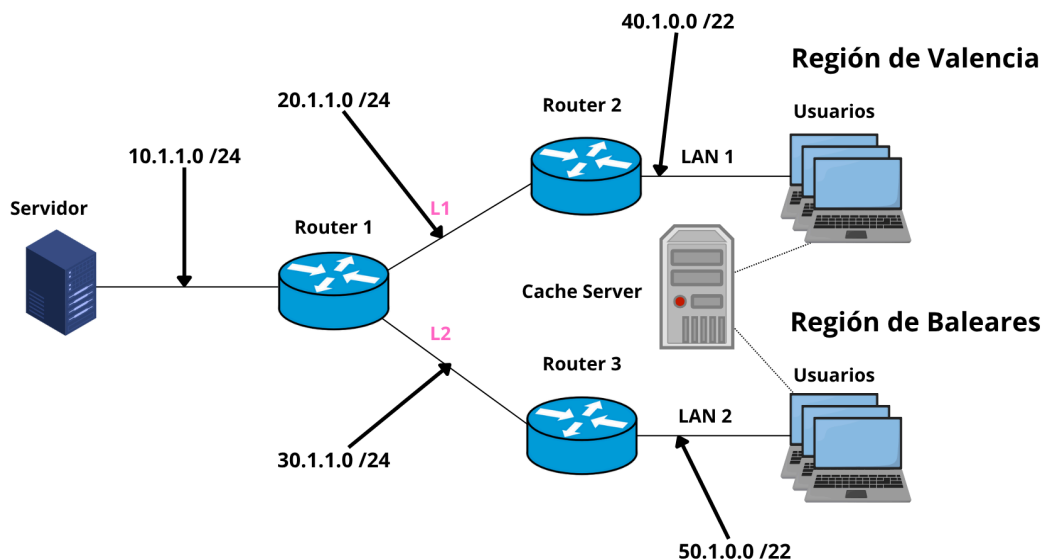
(Figura 6). Representa los enlaces y sus bitrates en el modelo

Cabe destacar que L1 y L2 son enlaces virtuales contratados al ISP (siendo L1 de especial interés ya que será donde se varíe el ancho de banda en función de los usuarios).

Los enlaces que aparecen en la figura conectados al servidor de caché son puramente representativos y no se van a tener en cuenta como elementos en ns3. El servidor caché, como se mencionó antes, se ha decidido modelar descartando un 60% del tráfico asumiendo que se consume en la propia LAN sin afectar al enlace.

6.1.3 Asignación de direcciones IP

Los rangos de direcciones que se usarán en las distintas subredes del modelo son los siguientes:



(Figura 6-2). Gráfica final bitrate/usuarios

Como se puede observar, para las subredes LAN se han usado máscaras /22 ya que el número de usuarios superaría las direcciones posibles si se eligiera /23 o /24.

6.2 Modelo de Tráfico y Simplificaciones

6.2.1 Tráfico de video y QoS

Las políticas de QoS del router no se van a tener en cuenta a la hora de hacer la simulación, ya que resulta demasiado complejo y hemos observado que no suponen un cambio significativo con respecto a los resultados esperados de la realidad, ya

que las políticas internas tendrían un impacto limitado si el problema radica en la capacidad total de dicho enlace externo.

6.2.2 Tráfico de usuarios

En nuestro sistema hablamos de **sesión de video**, que es el segmento temporal en el que el usuario empieza a demandar tráfico de video, es decir inicia sesión, hasta que se marcha.

Para poder modelar los datos de entrada del sistema necesitamos conocer los siguientes atributos:

- **Tiempo de servicio:** Es el tiempo de uso del servicio de cada usuario, es decir, el tiempo entre el inicio de sesión hasta su posterior salida.
- **Tiempo entre servicios:** Es el tiempo entre inicios de sesión consecutivos de un usuarios al sistema
- **Población:** Es el número de usuarios suscritos al servicio, el total de usuarios que pueden acceder al sistema

Para hacer el modelo estacionario vamos a hacer uso de la técnica llamada “hora cargada” que consiste en medir el tramo horario en el que mayor tráfico hay y obtener las medias. Esto son valores que la empresa ya nos suministra y son

- Tramo con mayor demanda: 10:00-11:00
- Promedio de la tasa de inicios de sesión por usuario: 0.6 IS(inicio sesion)/h

Con esto tenemos que para 100 usuarios que tiene el sistema, tenemos una tasa de inicio de sesión por usuario de 0.6 IS/h, suponiendo que la progresión es lineal iremos aumentando el número de usuarios para comprobar cuál es el ancho de banda conveniente a contratar sin perder calidad de servicio

Simplificaciones

A continuación expondremos las simplificaciones que hemos tenido en cuenta para hacer nuestro modelo:

- El porcentaje de población que consume las distintas calidades de vídeo suministradas (FullHD,HD y SD) se mantiene constante. Es respectivamente el 70%, 30%, 10%
- El tráfico de subida no se tendrá en cuenta.

A su vez en las sesiones de los usuarios se tendrán en cuenta las siguientes simplificaciones:

- Todos los vídeos de la sesión tendrán una duración fija y un bitrate fijo (según la calidad).
- El ancho de banda end-to-end es mayor al ancho de banda de codificación de video (consumo del video).

Estas simplificaciones de la sesión podemos verlas recogidas en [1] en el apartado 6 de dicho paper

Tiempo entre llegadas

Con las simplificaciones anteriores podemos definir la tasa de llegadas como una variable aleatoria discreta que sigue una distribución de poisson con parámetro λ , en nuestro caso todos los usuarios individualmente tienen $\lambda=0.6$ IS/h

Si el patrón de llegadas sigue una distribución de poisson, el tiempo entre dos llegadas consecutivas sigue una distribución exponencial. En nuestro caso será una variable aleatoria continua exponencial de parámetro $1/\lambda=6000s$

Por simplificación lo que haremos en la simulación es crear tantas aplicaciones generadoras de tráfico y receptoras como usuarios haya en la población del sistema.

Tiempo de servicio

Para caracterizar el tiempo de servicio en nuestra simulación es necesario analizar los datos estadísticos reales de la empresa para así poder comprobar si se puede modelar con alguna distribución.

Para definir el tiempo de servicio nos serviremos de [2] , en el que se expone que habiendo analizado una muestra de 4 millones de sesiones de video durante 3 meses, se ha podido comprobar que se puede modelar el tiempo de sesión como una variable aleatoria continua que sigue una distribución de weibull, y en dicho documento se comprueba que el 50% de la población muestran una distribución de weibull con el parámetro $k < 0$ y el otro 50% el parámetro es $k > 0$

Para tener esto en cuenta, en nuestra simulación lo que haremos será implementar el tiempo de servicio de la mitad de los usuarios con una variable aleatoria de weibull con parámetro de forma $k > 0$ y la otra mitad de los usuarios con parámetro $k < 0$

Con estos datos tendríamos ya terminado el modelo estadístico de los datos de entrada.

9. Diseño preliminar de los experimentos

En esta sección se definen las variables, constantes y métricas que guiarán la fase de experimentación. El objetivo es establecer un marco de trabajo claro para obtener los datos necesarios que permitan cumplir con los objetivos del proyecto.

9.1. Parámetros Invariables

Estos parámetros se mantendrán fijos en todas las simulaciones para asegurar una base de comparación coherente:

- **Bitrates de Tráfico de Vídeo:** Las tasas de bits para las distintas calidades de vídeo se fijan en 800 kbps (FullHD), 500 kbps (HD) y 200 kbps (SD).
- **Proporciones del Sistema:** Se mantendrá constante la distribución de usuarios por calidad (70% FullHD, 20% HD, 10% SD) y por región (70% Valencia, 30% Baleares).

9.2. Parámetros Configurables y de Control

Para facilitar la automatización de los experimentos, los siguientes parámetros del modelo son configurables mediante argumentos de línea de comandos:

- `num_usuarios`: Número total de usuarios en la simulación.
- `bitrate`: Ancho de banda a asignar al enlace L1 de Valencia.
- `semilla`: Semilla para el generador de números aleatorios, para garantizar la independencia estadística entre réplicas.

9.3. Factores de Estudio (Variables Dinámicas)

Los factores principales cuyo impacto se quiere analizar son:

- **Número de Usuarios:** Se estudiará un rango de 100 a 500 usuarios, en incrementos de 50, para simular el crecimiento de la plataforma.
- **Ancho de Banda del Enlace L1:** Para cada cohorte de usuarios, se planea realizar un barrido del ancho de banda en un **rango de 10 Mbps a 120 Mbps**. El objetivo de este barrido es encontrar el **valor mínimo que satisfaga los requisitos de QoS**, en lugar de analizar el rango completo.
- **Variables Aleatorias:** El comportamiento estocástico del sistema vendrá determinado por el **tiempo de servicio** (distribución de Weibull) y el **tiempo entre servicios** (distribución Exponencial) de cada usuario.

9.4. Medidas de Interés (Variables Dependientes)

Para evaluar la Calidad de Servicio (QoS) experimentada por los usuarios, se medirán las siguientes métricas al final de cada simulación, utilizando el módulo FlowMonitor:

- Porcentaje de Pérdida de Paquetes (%).
- Latencia Promedio por Flujo (ms).
- Jitter Promedio por Flujo (ms).

9.5. Planteamiento de la Gráfica de Resultados Objetivo

El entregable final de la experimentación será una gráfica que represente el **Ancho de Banda Mínimo Requerido** (eje Y) en función del **Número de Usuarios** (eje X), cumpliendo siempre con los criterios de QoS definidos en los objetivos.

10. Preparación de los datos de entrada

Este apartado describe el proceso de calibración del modelo, un paso crucial previo a la experimentación final. El objetivo fue determinar los valores de los parámetros internos del modelo (específicamente, los retardos de propagación de los enlaces) que no eran conocidos a priori. Para ello, se ajustó la simulación para que su comportamiento se correspondiera con un punto de operación conocido del sistema real de StreamNow.

10.1. Escenario de Calibración

Se utilizó como referencia el único dato de rendimiento del sistema real proporcionado por la empresa, que se detalla en la fase de validación:

- **Carga de Usuarios:** 100
- **Ancho de Banda (Enlace L1):** 8 Mbps
- **Resultado Real (Métrica Objetivo):** Latencia media de 5 ms

10.2. Proceso Iterativo de Calibración de Retardos

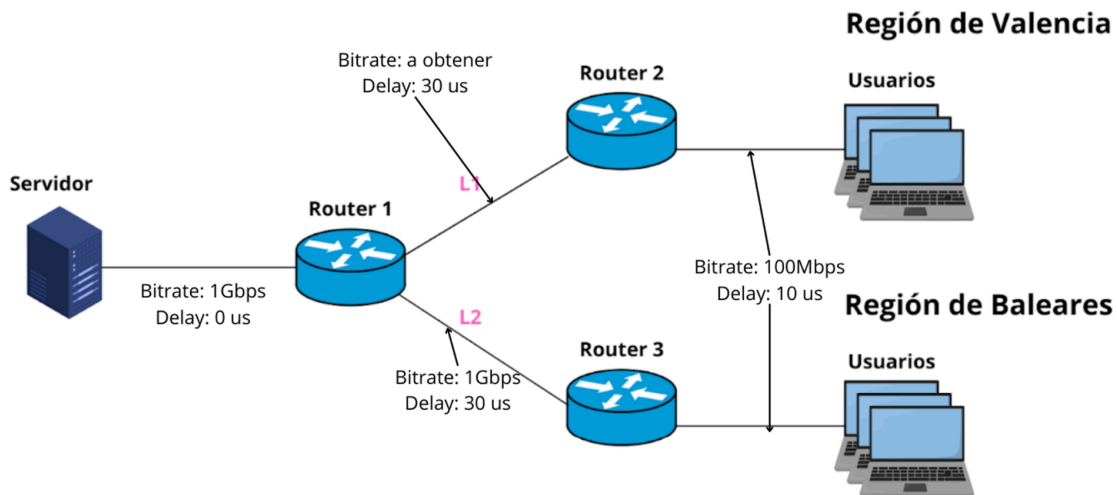
El proceso para encontrar los retardos de enlace adecuados fue iterativo. Se asignaron valores iniciales a los retardos de los enlaces en el script de simulación y se ejecutaron varias réplicas bajo el escenario de calibración, comparando la latencia media obtenida con el valor objetivo de 5 ms. Los retardos se ajustaron en sucesivas iteraciones hasta que la salida del modelo se estabilizó en un valor cercano y coherente con la referencia real.

10.3. Valores de Retardo Finales

Al concluir este proceso de calibración, se establecieron los siguientes valores de retardo de propagación. Estos valores se consideraron parte del modelo ya calibrado y, por tanto, se mantuvieron fijos como constantes para todos los experimentos posteriores:

- **Enlace Servidor ↔ Router 1:** 0 μ s
- **Enlaces Troncales L1 y L2:** 30 μ s
- **Enlaces LAN de Acceso (a Usuarios):** 10 μ s

Estos valores finales se representan en la topología del modelo en la Figura 7.

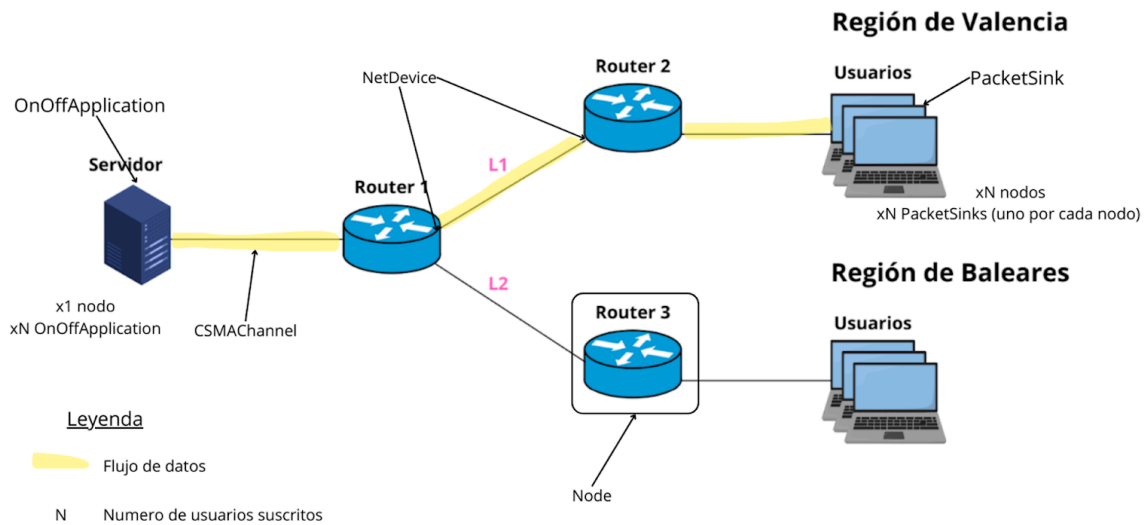


(Figura 7). Representa bitrate y delay de los enlaces del modelo

11. Implementación del modelo

Para llevar a cabo la simulación, el modelo especificado en el apartado 6 se ha traducido a un script de C++ utilizando las librerías del simulador ns-3. Este apartado detalla los componentes de ns-3 seleccionados y su configuración para replicar la topología, el comportamiento del tráfico y las condiciones de la red de StreamNow.

11.1. Identificación y creación de los elementos de red

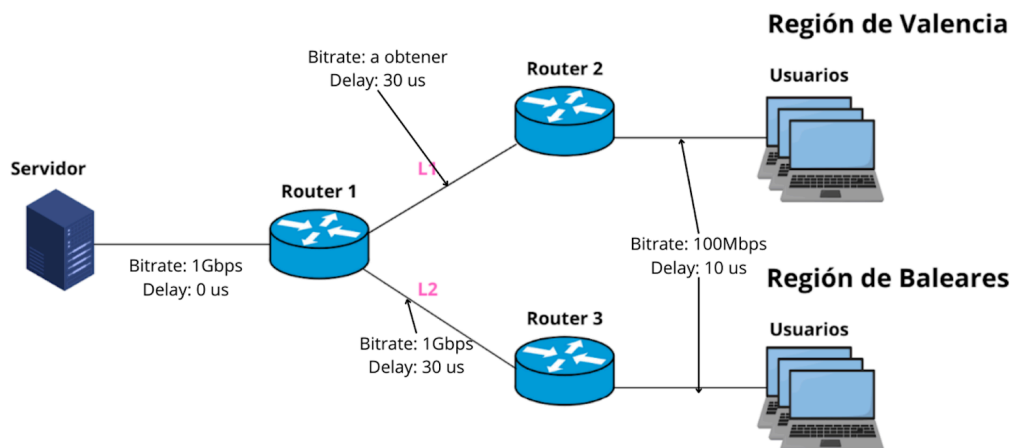


(Figura 8). Elementos ns3 implicados en el modelo

La arquitectura de la red se construye utilizando los siguientes elementos fundamentales de ns-3:

- **Nodos (ns3::NodeContainer):** Se crean contenedores de nodos para representar todos los equipos de la red. La distribución es la siguiente:
 - Un único nodo para el servidor central.
 - Un contenedor con 3 nodos que actúan como los routers de la red.
 - Dos contenedores de nodos para los usuarios finales, uno para la región de Valencia y otro para Baleares. El número de usuarios en cada región se calcula dinámicamente en función del total de usuarios de la simulación y las proporciones definidas (70% en Valencia, 30% en Baleares).
- **Pila de Protocolos y Enrutamiento (ns3::InternetStackHelper, ns3::RipHelper):** Se instala una pila de protocolos TCP/IP en todos los nodos para habilitar la comunicación de red. Para el enrutamiento, y como mejora sobre el planteamiento inicial de rutas estáticas, se ha implementado el protocolo de enrutamiento dinámico **RIP** (Routing Information Protocol). Esto se logra mediante la clase ns3::Ipv4ListRoutingHelper, que permite combinar distintos protocolos, asignando una prioridad más alta a RIP sobre las rutas estáticas. Esto dota al modelo de mayor realismo y robustez.

- **Enlaces de Red (ns3::CsmaHelper):** Todos los enlaces de la red se modelan como canales CSMA (Carrier Sense Multiple Access), que simulan el comportamiento de redes Ethernet.
 - Se utilizan distintas instancias de CsmaHelper para configurar los enlaces troncales (1 Gbps o bitrate variable para L1) y los enlaces de acceso a usuarios (100 Mbps).
 - Cada conexión genera un NetDeviceContainer, que representa las tarjetas de red virtuales en los nodos.
- **Enlaces de Red (ns3::CsmaHelper):** Todos los enlaces de la red se modelan como canales CSMA (Carrier Sense Multiple Access) para simular el comportamiento de Ethernet (IEEE 802.3bj). La configuración de cada enlace es la siguiente:
 - **Velocidad:** La velocidad de los enlaces es fija (1 Gbps para el troncal principal y el enlace a Baleares; 100 Mbps para las LANs de acceso), a excepción del enlace L1 (hacia Valencia), cuyo ancho de banda es una variable del estudio.
 - **Retardo:** Los valores de retardo de propagación, determinados durante la fase de calibración (apartado 10.2), se establecieron en 0 μ s (enlace Servidor-Router1), 30 μ s (enlaces troncales L1 y L2) y 10 μ s (enlaces de acceso a usuarios). Estos valores se mantuvieron fijos durante toda la experimentación.



(Figura 9). Bitrate y delay de enlaces del modelo

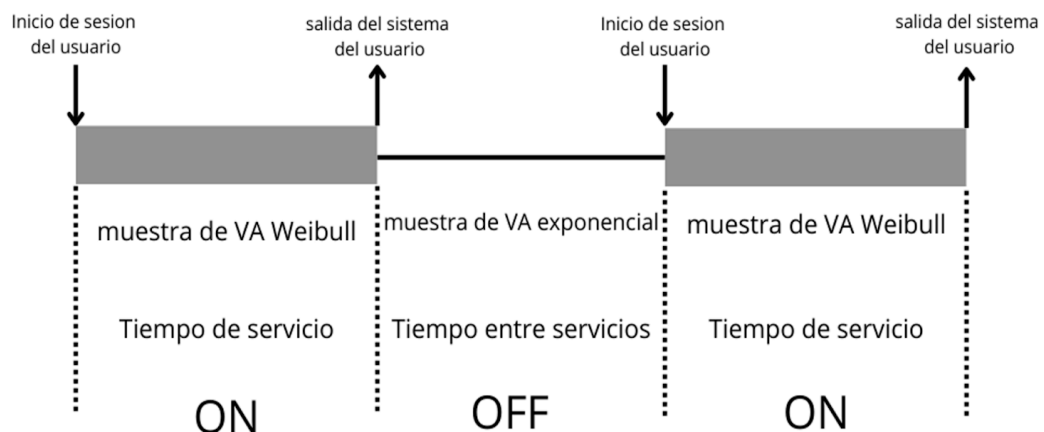
- **Direccionamiento IP (ns3::Ipv4AddressHelper):** Se asignan direcciones IP únicas a cada interfaz de red. Las subredes definidas en la simulación son:
 - 10.1.1.0/24: Enlace Servidor \leftrightarrow Router 1.
 - 20.1.1.0/24: Enlace L1 (Router 1 \leftrightarrow Router 2, Valencia).
 - 30.1.1.0/24: Enlace L2 (Router 1 \leftrightarrow Router 3, Baleares).
 - 40.1.0.0/22: LAN de Valencia (Router 2 \leftrightarrow Usuarios Valencia).

- 50.1.0.0/22: LAN de Baleares (Router 3 ↔ Usuarios Baleares).

11.2. Configuración del Tráfico y las Aplicaciones

El tráfico de streaming de vídeo se modela mediante un par de aplicaciones cliente-servidor que simulan el patrón de consumo de los usuarios sobre TCP.

- **Generador de Tráfico (ns3::OnOffHelper):** En el nodo servidor se instalan múltiples aplicaciones ns3::OnOffApplication utilizando el ayudante OnOffHelper. Cada aplicación se configura para usar sockets TCP ("ns3::TcpSocketFactory") y genera tráfico hacia un usuario concreto, simulando una sesión de streaming.
- **Patrón ON/OFF:** El comportamiento de cada usuario se modela con los parámetros OnTime y OffTime:
 - **OnTime (Tiempo de servicio):** Sigue una distribución de **Weibull** para modelar la duración de la sesión de vídeo. Para reflejar la variabilidad en el comportamiento de los usuarios, se alternan dos parametrizaciones: la mitad de los flujos usan una forma $k=1.1$ y la otra mitad $k=0.9$, ambas con una escala de 300 segundos. Esto se implementa con la clase ns3::WeibullRandomVariable.
 - **OffTime (Tiempo entre servicios):** Sigue una distribución **Exponencial** con una media de 6000 segundos ($1/\lambda$) para modelar el tiempo que pasa hasta que un usuario inicia una nueva sesión. Esto se implementa con ns3::ExponentialRandomVariable.



(Figura 10). Patrón OnOff

- **Tasa de Bits y Calidad de Vídeo:** La tasa de bits del tráfico generado durante el estado "ON" es constante y depende de la calidad del vídeo asignada a ese flujo (SetConstantRate):
 - FullHD: 800 kb/s
 - HD: 500 kb/s
 - SD: 200 kb/s
- **Simulación de Caché:** Para aumentar el realismo, se ha modelado un **ratio de acierto de caché del 60%**. Esto significa que solo el 40% de las solicitudes de los usuarios generan tráfico desde el servidor central. En el código, esto se consigue creando aplicaciones OnOffApplication únicamente para el 40% de los usuarios de cada categoría ($\text{numValFHD} * 0.4$), reduciendo la carga en la red troncal.
- **Receptor de Tráfico (ns3::PacketSinkHelper):** En cada uno de los nodos de usuario se instala una aplicación ns3::PacketSink que recibe el tráfico enviado por el servidor, lo consume y permite al FlowMonitor realizar mediciones sobre el flujo. Se usan puertos distintos para cada región (puerto 9 para Valencia, puerto 10 para Baleares).

11.3 Configuración y Control de la Simulación

El entorno de simulación se configura para asegurar la replicabilidad y la correcta recolección de datos.

- **Duración de la Simulación:** Cada ejecución individual tiene una duración fija de **45 segundos** de tiempo simulado (`Simulator::Stop(Seconds(45.0))`). Esta duración, aunque corta, es suficiente por diseño:
 - Las aplicaciones comienzan a generar tráfico a los **10 segundos**, dando tiempo al protocolo RIP para converger y estabilizar las tablas de enrutamiento.
 - Las aplicaciones se detienen a los **40 segundos**, asegurando que se complete al menos un ciclo "ON" (cuya media es de 300s, pero la aplicación solo dura 30s en este caso) donde se pueda medir el rendimiento en estado estacionario.
 - Los 5 segundos finales (de 40s a 45s) actúan como un búfer para asegurar que los últimos paquetes en tránsito lleguen a su destino y sean contabilizados.
- **Semilla Aleatoria:** Para garantizar la independencia estadística entre las 10 réplicas de cada experimento, la semilla del generador de números aleatorios

se establece al inicio de cada ejecución mediante `RngSeedManager::SetSeed(semilla)`. El script `run.sh` se encarga de proporcionar una semilla única para cada réplica.

11.4. Medición de Métricas de Interés

La recolección de resultados se centraliza a través del módulo `FlowMonitor`, que proporciona un análisis detallado de cada flujo de datos en la simulación.

- **Instalación (`ns3::FlowMonitorHelper`):** Se instala una instancia de `FlowMonitor` en todos los nodos, que captura cada paquete transmitido y recibido, agrupándolos por flujo (`flowmonHelper.InstallAll()`).
- **Procesamiento de Resultados:** Al finalizar la simulación (`Simulator::Run()`), el código itera sobre las estadísticas de todos los flujos (`flowmon->GetFlowStats()`). Se descartan los flujos con menos de 10 paquetes para evitar datos anómalos de flujos de control (`if (flowStats.txPackets > 10)`). Para los flujos válidos, se calculan las siguientes métricas:
 - **Pérdida de Paquetes (%):** Cociente entre el total de paquetes perdidos (`flowStats.lostPackets`) y el total de paquetes enviados (`flowStats.txPackets`), expresado en porcentaje.
 - **Latencia Media (ms):** Para cada flujo, se obtiene la latencia media (`flowStats.delaySum / flowStats.rxPackets`). El resultado final es la media de las latencias de todos los flujos, calculada con la clase `ns3::Average<double>`.
 - **Jitter Medio (ms):** De forma análoga, se obtiene el jitter medio de cada flujo (`flowStats.jitterSum / (flowStats.rxPackets - 1)`) y se promedia con `ns3::Average<double>`.
- **Salida de Datos:** Los resultados finales de cada ejecución (n° de usuarios, bitrate y las tres métricas) se escriben en una nueva línea en el fichero `results.dat` en modo apendicitis (`std::ios::app`). Este fichero está formateado para servir como entrada directa al script de análisis `plot-results.cc`, completando el flujo de trabajo automatizado.

11.5. Arquitectura del programa de simulación

Para poder visualizar la consecución de actividades realizadas por nuestro código nos serviremos de un diagrama de actividad, nuestro código contiene una única clase main que realiza todas las funciones citadas a continuación.

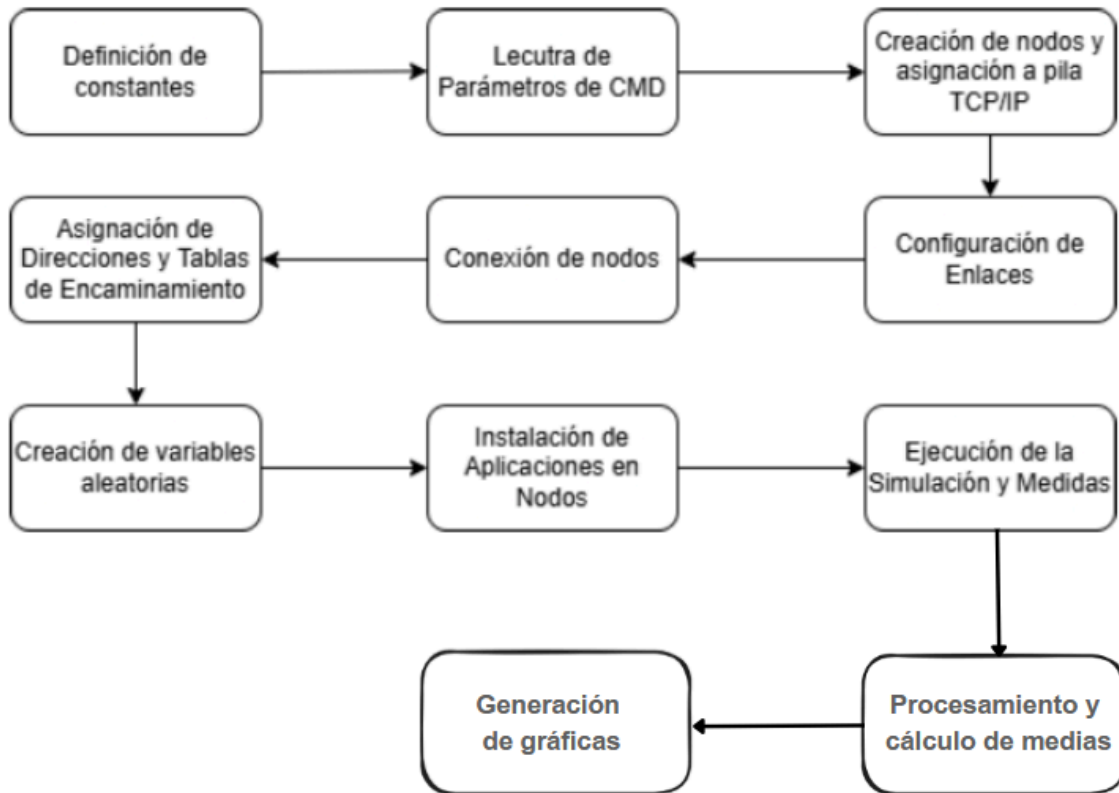


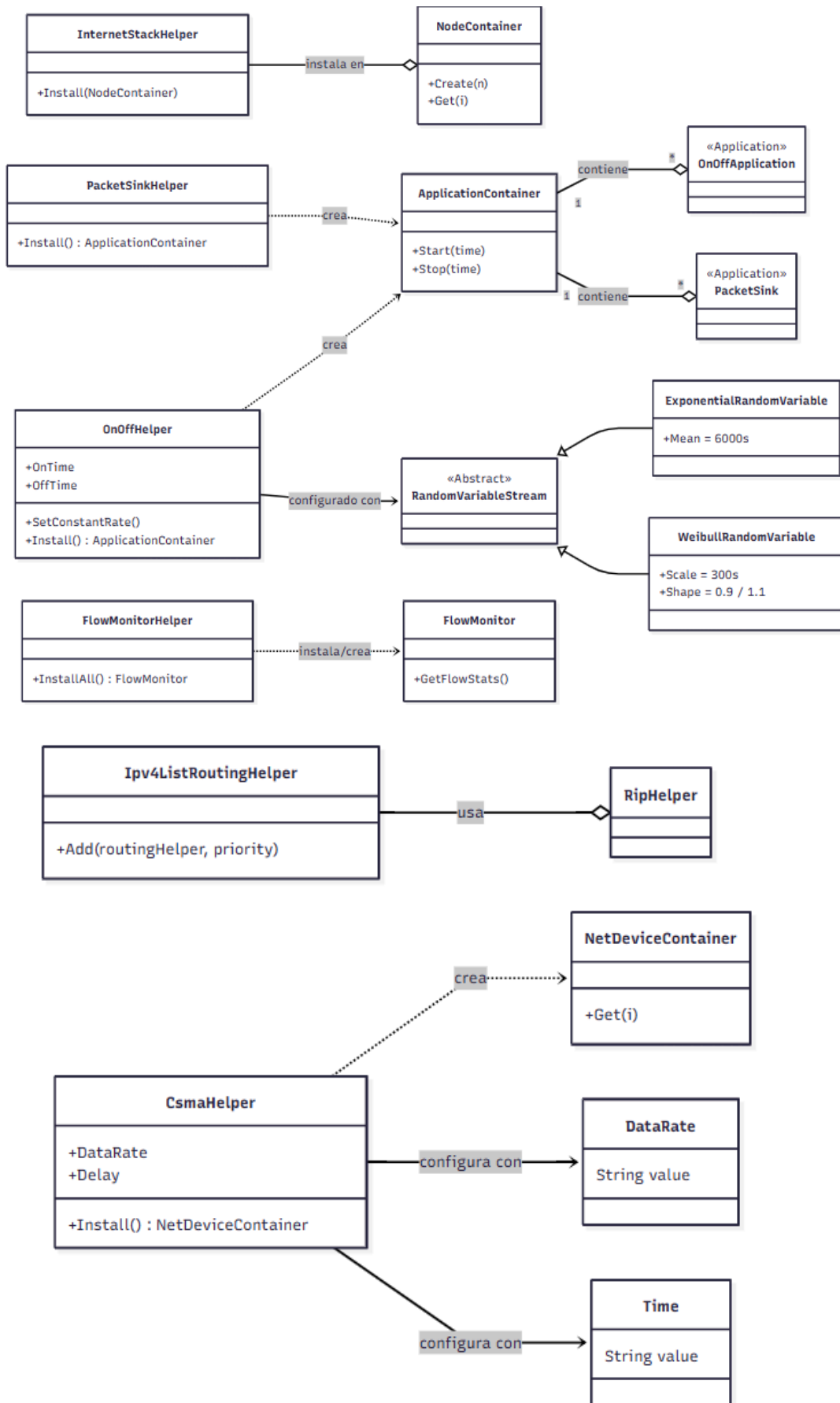
Figura 11 (diagrama de flujo)

A continuación nombraremos las clases definidas de NS3 (es decir las que ns3 facilita para su uso) que se usan en cada paso del diagrama

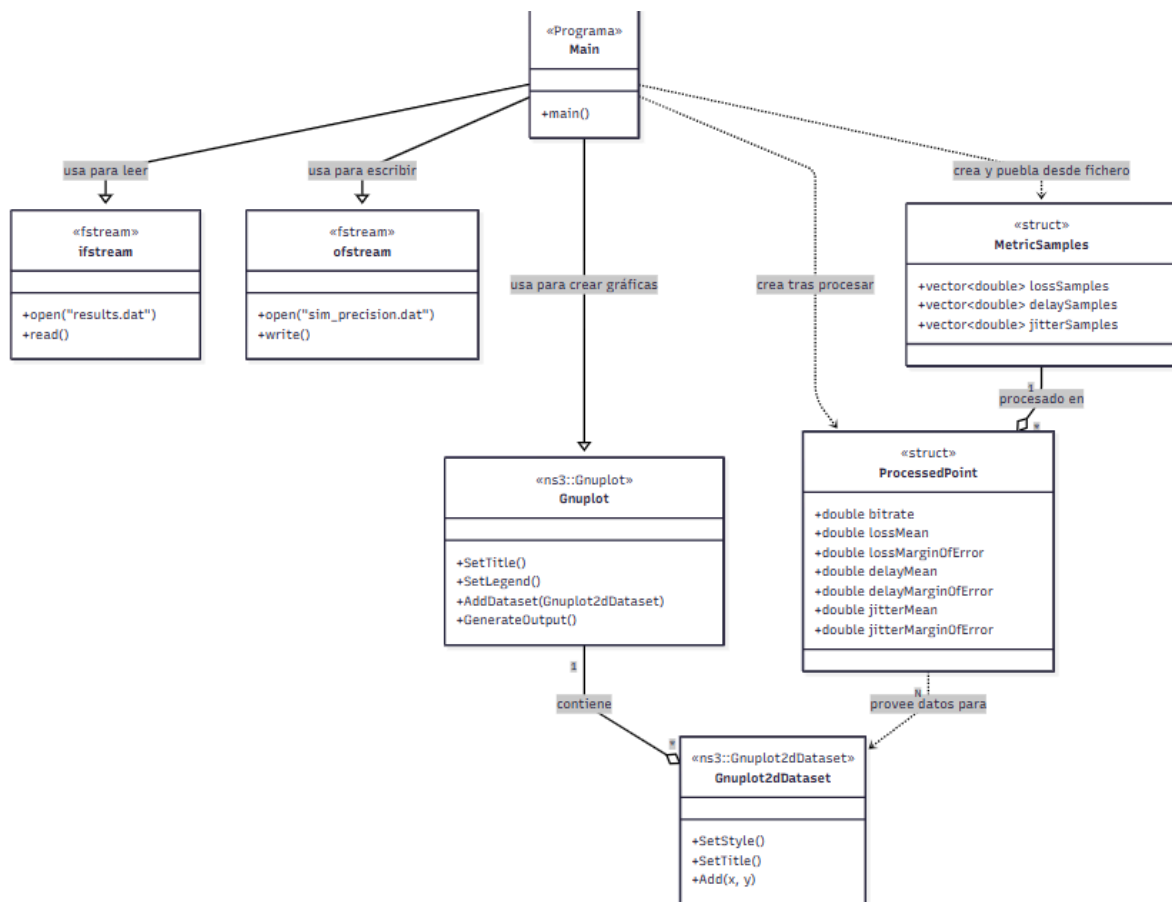
1. **Definición de constantes:** Aquí definimos los valores fijos de nuestra simulación
2. **Lectura de parametros CMD:** Aquí establecemos las variables que queremos introducir en nuestra simulación mediante comandos con la clase *CommandLine*, en nuestro caso tenemos: num usuarios y semilla
3. **Creación de nodos y pilas:** Se crean nodos para el servidor, routers y usuarios (*NodeContainer*), y se instala la pila TCP/IP (*InternetStackHelper*) con enrutamiento RIP (*Ipv4ListRoutingHelper*, *RipHelper*).

- 4. Configuración de enlaces:** Creamos los enlaces que vamos a utilizar usando la clase *CsmaHelper* que crea tanto un *NetDevice* (tarjeta de red) como un *CsmaChannel* (Cable)
- 5. Conexión de nodos:** Con la clase *CsmaHelper* instalamos los enlaces según nuestra topología, nos devuelve una instancia de *Ipv4InterfaceContainer*
- 6. Asignación de direcciones y tabla de encaminamiento:** Se asignan direcciones IP (*Ipv4AddressHelper*) y se configuran las tablas de enrutamiento
- 7. Creación de variables aleatorias:** Para la generación de variables aleatorias utilizamos las clases *ExponentialRandomVariable* y *WeibullRandomVariable*.
- 8. Instalación de aplicaciones en nodos:** *IPacketSink* en usuarios y *OnOffApplication* en el servidor (simulando 40% de tráfico, 60% caché) con tasas de bits por calidad
- 9. Ejecución de la simulación y medidas:** La simulación corre por 45 segundos, *FlowMonitor* recolecta métricas (pérdida de paquetes, latencia, jitter), y los resultados se guardan en *results.dat*.
- 10. Procesamiento y cálculo de medias:** Se calculan medias y márgenes de error (IC 95%) para latencia, jitter y pérdida de paquetes a partir de las 10 réplicas obtenidas para cada punto de simulación.
- 11. Generación de gráficas:** Usando los datos procesados, *plot-results.cc* genera gráficas detalladas (retardo, jitter, pérdida por bitrate), de precisión, y la final de bitrate requerido vs. usuarios, con *Gnuplot*.

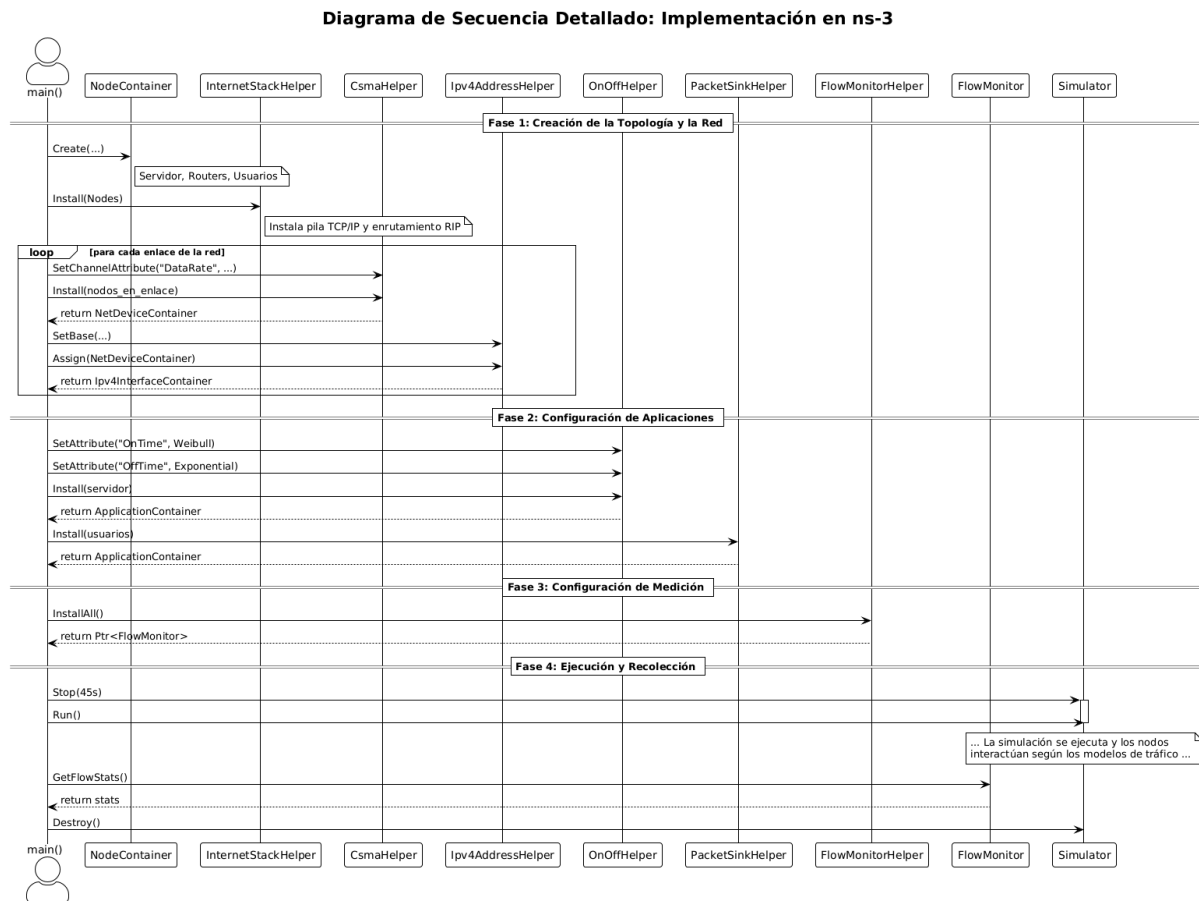
En las páginas siguientes tenemos los diagramas de clases y secuencia de nuestro programa:



(Figura 12). Clases implicadas en el código principal `onoffRoutin.cc`

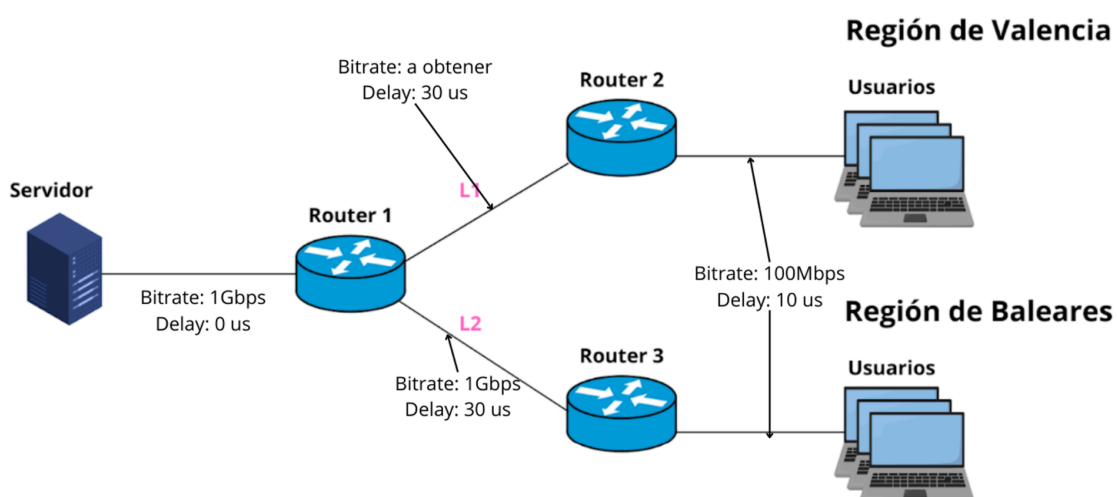


(Figura 13). Diagrama de clases del programa `plot-results.cc`



(Figura 14). Diagrama de secuencia del programa

Aquí podemos ver los valores finales de configuración del retardo de cada enlace, después de validar el programa:



(Figura 15). Valores finales de retardo en los enlaces del modelo

12. Verificación y validación

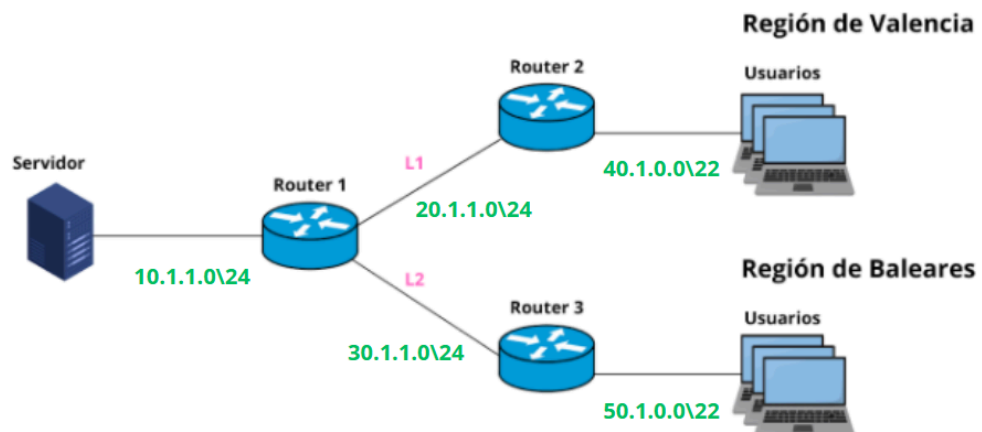
12.1. Verificación

El proceso de verificación se llevó a cabo de forma incremental durante todo el desarrollo para asegurar que el modelo implementado en ns-3 (onoffRouting.cc) fuera una traducción correcta y libre de errores del modelo conceptual. Se aplicaron diversas técnicas para depurar y confirmar el comportamiento de cada componente del sistema, desde la topología de red hasta la lógica de la recolección de datos.

12.1.1. Verificación de la Topología y el Enrutamiento

La primera fase se centró en asegurar que la estructura de la red era correcta y que los paquetes podían ser encaminados de un extremo a otro.

- **Asignación de Direcciones IP:** Tras la configuración de las subredes mediante Ipv4AddressHelper, se utilizó el sistema de registros de ns-3 (NS_LOG_DEBUG) para imprimir en pantalla las direcciones IP asignadas a cada NetDevice de los nodos. Se comprobó que cada interfaz (servidor, routers y usuarios) tenía la dirección IP y la máscara de subred esperadas según el diseño.



(Figura 16). Asignación de direcciones IP

- **Conectividad y Enrutamiento Dinámico (RIP):** Para verificar el correcto funcionamiento de RIP, se realizaron dos comprobaciones clave:

1. **Convergencia de Rutas:** Se habilitó el logging específico para el protocolo RIP (NS_LOG_INFO="Rip"). Al ejecutar la simulación, se observó en los logs cómo los routers intercambiaban sus tablas de enrutamiento durante los primeros segundos.
2. **Inspección de Tablas de Enrutamiento:** Se insertó código para que, pasados los 10 segundos iniciales de convergencia, se imprimieran las tablas de enrutamiento IPv4 de los tres routers. Se verificó que el **Router 1** había aprendido las rutas hacia las LAN de Valencia (40.1.0.0/22) y Baleares (50.1.0.0/22), y que los **Routers 2 y 3** conocían la ruta hacia el servidor (10.1.1.0/24).

12.1.2. Verificación del Modelo de Tráfico

Una vez confirmada la topología, se verificó que el comportamiento de las aplicaciones que generaban el tráfico de vídeo se ajustaba a los modelos estadísticos definidos.

- **Generación de Aplicaciones (Simulación de Caché):** El código está diseñado para crear aplicaciones OnOffApplication solo para el 40% de los usuarios, simulando el efecto de la caché. Se añadió una traza de log (NS_LOG_INFO) para contar el número de aplicaciones generadas para cada tipo de calidad de vídeo. Se confirmó que, para N usuarios totales, el número de aplicaciones siempre correspondía a la fórmula ($N * \%Region * \%Calidad * 0.4$), validando la correcta implementación de la lógica de la caché.
- **Patrón ON-OFF y Variables Aleatorias:** Verificar el comportamiento de variables aleatorias es complejo. Para ello, se creó una versión de prueba del script donde se reemplazaron las WeibullRandomVariable y ExponentialRandomVariable por variables constantes (ConstantRandomVariable).
 - **Test:** Se configuró OnTime con un valor fijo de 2 segundos y OffTime con 5 segundos.
 - **Medición:** Se habilitó la exportación de trazas de la simulación a ficheros .pcap (CsmaHelper::EnablePcapAll).
 - **Análisis:** Estos ficheros .pcap se analizaron con la herramienta **Wireshark**. Se observó claramente que el tráfico entre el servidor y un usuario consistía en ráfagas de paquetes durante exactamente 2 segundos, seguidas de un silencio absoluto de 5 segundos. Este ciclo se repetía, confirmando que la lógica ON-OFF funcionaba como se esperaba.

12.1.3. Verificación del Flujo Experimental y la Recolección de Datos

Finalmente, se verificó que todo el sistema automatizado, desde la ejecución de experimentos hasta el análisis de resultados, funcionaba correctamente.

- **Paso de Parámetros (run.sh a onoffRouting.cc):** Al final de cada ejecución de onoffRouting.cc, una línea de log imprime un resumen de los resultados y los parámetros de entrada (NS_LOG_INFO("Users: " << num_usuarios << ", Bitrate: " << bitrate_str ...)). Se revisó esta salida para confirmar que los valores de num_usuarios, bitrate y semilla pasados por el script run.sh eran recibidos y utilizados correctamente por la simulación.
- **Integridad de los Datos de Salida:** Se inspeccionó el fichero results.dat tras una ejecución de prueba. Se verificó que el script run.sh generaba el número correcto de líneas de resultados: para U valores de usuarios y B valores de bitrate, el fichero contenía $U * B * 10$ líneas, confirmando que el bucle de 10 réplicas funcionaba.
- **Procesamiento de Resultados (plot-results.cc):** Se creó un fichero results.dat de prueba con datos controlados (por ejemplo, 10 réplicas con un valor de latencia de 15ms). Al ejecutar el script plot-results.cc con este fichero, se comprobó que el fichero de salida sim_precision.dat mostraba una media de 15.0 y un margen de error de 0.0, validando la correctitud de los cálculos estadísticos.

12.2. Validación

El objetivo de la validación es evaluar la representatividad del modelo de simulación comparándolo con un punto de operación conocido del sistema real de StreamNow, para así generar confianza en los resultados del proyecto.

12.2.1. Objetivo y Metodología de Validación

Dado que los experimentos finales se ejecutaron en un barrido de 10 a 120 Mbps, la validación se realizará utilizando el escenario simulado más cercano: **100 usuarios con un ancho de banda de 10 Mbps**, cuyos datos se extraen de los ficheros de resultados.

12.2.2. Resultados del Experimento de Validación

Se analizaron las 10 réplicas del escenario de 100 usuarios y 10 Mbps. Los resultados estadísticos para la latencia se resumen en la siguiente tabla:

Métrica	Valor
Media Muestral (\bar{x})	61.05 ms
Margen de Error (95%)	± 2.93 ms
Intervalo de Confianza (95%)	[58.12, 63.98] ms

(Tabla 7). Resultados estadísticos

Como se observa, los resultados son consistentes y presentan una variabilidad baja, lo que indica un comportamiento estable del modelo bajo estas condiciones de alta congestión.

12.2.3. Interpretación y Conclusiones de la Validación

El resultado de la simulación (~61 ms) es significativamente mayor que el valor de referencia real (5 ms). Esta notable discrepancia no invalida el modelo, sino que, por el contrario, **lo valida de una forma más profunda y revela un hallazgo clave**:

1. **El Modelo Predice Correctamente la Congestión:** El modelo demuestra que un enlace de 10 Mbps es claramente insuficiente para 100 usuarios, resultando en una alta latencia y una pérdida de paquetes superior al 0.5%. Este es un comportamiento de red físicamente realista.
2. **La Discrepancia Resalta el Impacto de las Simplificaciones:** La capacidad del sistema real de operar a 5 ms con solo 10 Mbps sugiere que su rendimiento depende de factores no modelados. La diferencia se atribuye a las **sofisticadas políticas de Calidad de Servicio (QoS)** (como DSCP, WRED, etc.) que se abstrajeron del modelo para centrar el estudio en el dimensionamiento del ancho de banda bruto.

Por tanto, se considera el **modelo validado**. No porque replique numéricamente el punto de referencia (lo cual es imposible debido a las simplificaciones realizadas), sino porque su comportamiento ante la congestión es coherente y la desviación respecto a la realidad expone y justifica el impacto de dichas simplificaciones.

13. Diseño final de los experimentos

Esta sección detalla el diseño definitivo de los experimentos a realizar en ns-3, estableciendo los parámetros, las métricas y la metodología para obtener los resultados que permitan cumplir con los objetivos del proyecto.

13.1. Consideraciones Generales de la Simulación

Para asegurar la fiabilidad de los resultados, se llevarán a cabo 10 simulaciones (réplicas) por cada combinación de número de usuarios y ancho de banda del enlace L1. Posteriormente, se calculará la media de estas réplicas para obtener un valor representativo de las métricas de interés.

La condición de fin de la simulación para cada réplica se establece en 45 segundos de tiempo simulado (**Simulator::Stop(Seconds(45.0))**). Esta duración se ha determinado cuidadosamente:

- Las aplicaciones de tráfico (**OnOffApplication**) comienzan su actividad a los 10 segundos, lo que proporciona un tiempo suficiente para que el protocolo de enrutamiento RIP converja y las tablas de enrutamiento se estabilicen en la red.
- Las aplicaciones se detienen a los 40 segundos. Aunque la duración media de una sesión ON de vídeo es de 300 segundos, este intervalo de 30 segundos de actividad es suficiente para observar el comportamiento del tráfico en estado estacionario y recoger las métricas relevantes.
- Los 5 segundos finales (de 40s a 45s) actúan como un búfer para asegurar que todos los paquetes generados en los flujos tengan tiempo de llegar a su destino y sean contabilizados por el módulo FlowMonitor.

13.2. Gráficas Esperadas

El entregable final del proyecto será una gráfica principal, acompañada de gráficas de detalle para cada métrica de QoS.

13.2.1. Gráfica Principal: Bitrate Mínimo Requerido vs. Número de Usuarios

- **Eje de Abscisas (Eje X): "Número de Usuarios".**
 - Unidades: Usuarios.

- Valores: Desde 100 hasta 400 usuarios (dado que para 450 y 500 usuarios no se encontró solución viable con la infraestructura actual), en saltos de 50 en 50.
- Número de puntos: 7 puntos (100, 150, 200, 250, 300, 350, 400).
- **Eje de Ordenadas (Eje Y): "Bitrate Mínimo Requerido".**
 - Unidades: Mbps.
 - Valores: Dependerán de los resultados de la simulación, oscilando entre 30 Mbps y 100 Mbps, en saltos de 10 en 10.
- **Curvas:** Una única curva que represente el "Bitrate Requerido" para cumplir con los requisitos de QoS.

Esta gráfica la encontramos en la Figura 17

13.2.2. Gráficas de Métricas de QoS (Retardo, Jitter, Pérdida de Paquetes) vs. Bitrate

Para cada una de las métricas (Retardo Medio, Jitter Medio, Pérdida de Paquetes), se generará una gráfica similar a las Figuras 18, 19 y 20 del documento.

- **Eje de Abscisas (Eje X): "Bitrate".**
 - Unidades: Mbps.
 - Valores: Desde 10 Mbps hasta 120 Mbps.
 - Número de puntos: 12 puntos (10, 20, ..., 120).
- **Eje de Ordenadas (Eje Y):**
 - "Retardo Medio" (ms).
 - "Jitter Medio" (ms).
 - "Pérdida de Paquetes" (porcentaje).
- **Curvas:** Múltiples curvas, una por cada conjunto de usuarios simulado.
 - Parámetro de la curva: "Número de Usuarios".
 - Valores inicial y final: De 100 a 500 usuarios.
 - Número de curvas: 9 curvas (100, 150, 200, 250, 300, 350, 400, 450, 500 usuarios).
 - Cada gráfica incluirá una línea de "Umbral" para indicar el valor máximo aceptable de la métrica (20 ms para retardo, 8 ms para jitter, 0.5% para pérdida de paquetes).

14. Experimentación

Para la obtención de los siguientes datos y gráficas se ha realizado un proceso de simulación en una máquina virtual Ubuntu proporcionada por la Universidad de Sevilla que ha durado en total 40 horas.

Esta duración tan prolongada es debida a que hemos realizado 10 simulaciones por cada muestra de usuarios-bitrate, calculando posteriormente la media de los resultados, obteniendo así gran robustez en los resultados obtenidos a cambio de un tiempo de simulación elevado debido a las limitaciones de recursos hardware.

En concreto, hemos variado los usuarios de 100 a 500 con saltos de 50 y el bitrate de 10 a 120 Mbps con saltos de 10. Por tanto el total de muestras es de 88, y el total de simulaciones es de 880.

Procedimiento para la obtención de los datos y gráficas

Para generar los resultados de este proyecto se ha seguido un flujo de trabajo automatizado, basado en tres programas principales que interactúan entre sí.

Estructura de ficheros

Para entender bien el procedimiento realizado, hay que recordar que se dispone de 3 programas principales: run.sh (script en Bash que orquesta toda la ejecución, en él se pueden definir los rangos de usuario y bitrate a simular, entre otras cosas), onoffRouting.cc (programa de ns-3 que contiene el escenario del modelo y ejecuta las simulaciones, generando resultados en forma de tuplas de retardo, jitter y pérdida de paquetes) y plot-results.cc (programa de ns-3 encargado de procesar los resultados y generar las gráficas finales a través de Gnuplot).

Flujo de datos

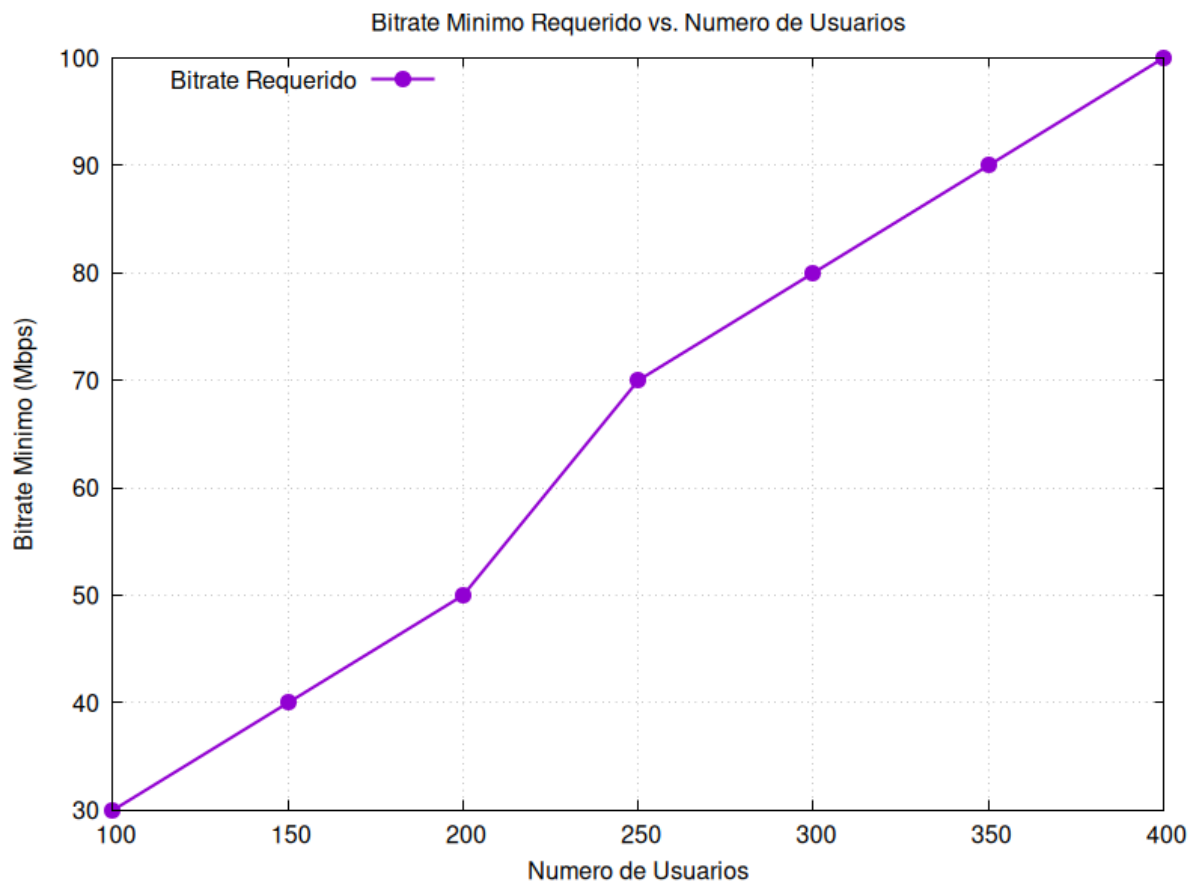
Al ejecutar run.sh, este ejecuta las simulaciones a través de onoffRouting.cc, generando un fichero llamado results.dat donde se almacenan los datos de cada simulación en tuplas de [usuarios, bitrate, perdida-paquetes, retardo, jitter].

A continuación, run.sh ejecuta el programa plot-results.cc, que realiza dos tareas:

1. Primero, genera un fichero sim_precision.dat donde se guardan los datos estadísticos, con tuplas de [usuarios, bitrate, latenciaMedia, latenciaError95, jitterMedio, jitterError95, perdidaMedia, perdidaError95].
2. Después, genera todas las gráficas: las de la solución final y las métricas se guardan en un directorio llamado "graficas", mientras que las de los márgenes de error se guardan en un directorio llamado "graficas-precision".

Gráfica objetivo final

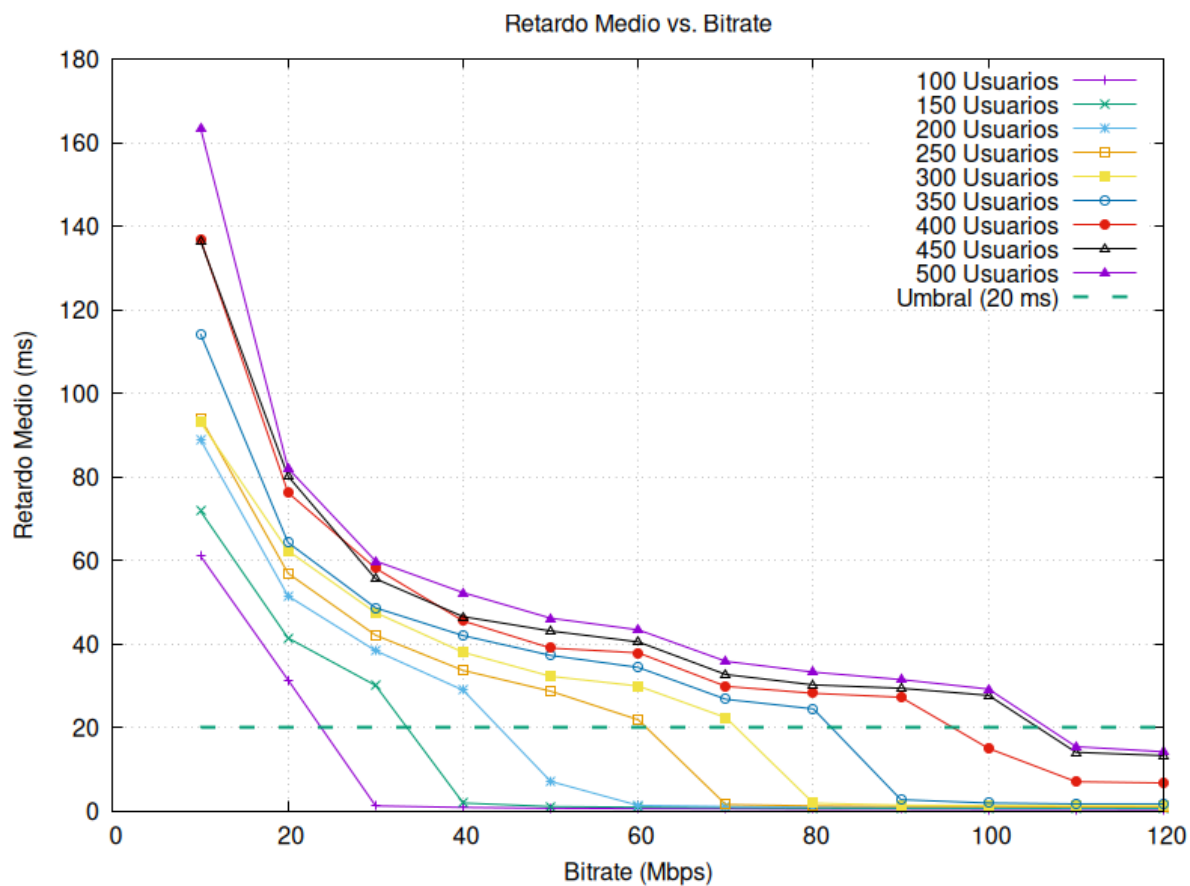
La gráfica final que representa el bitrate requerido para cada rango de usuarios cumpliendo los requisitos mínimos exigidos es la siguiente:



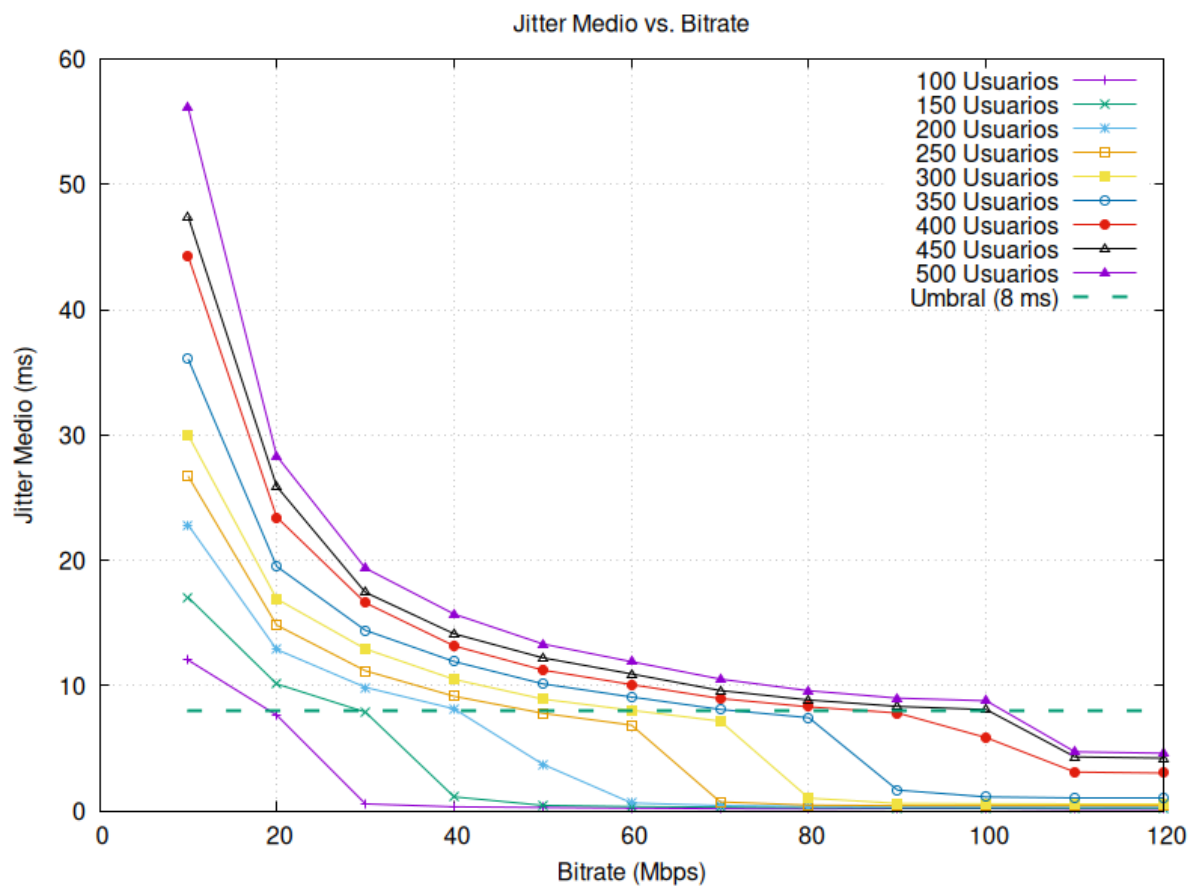
(Figura 17) Gráfica final bitrate/usuarios

Nota: Para los rangos de 450 y 500 usuarios no se ha encontrado solución, es por eso que no se han representado los puntos en la gráfica. Esto se debe a un fenómeno que se estudiará en detalle en el apartado de "Análisis e Interpretación de los resultados".

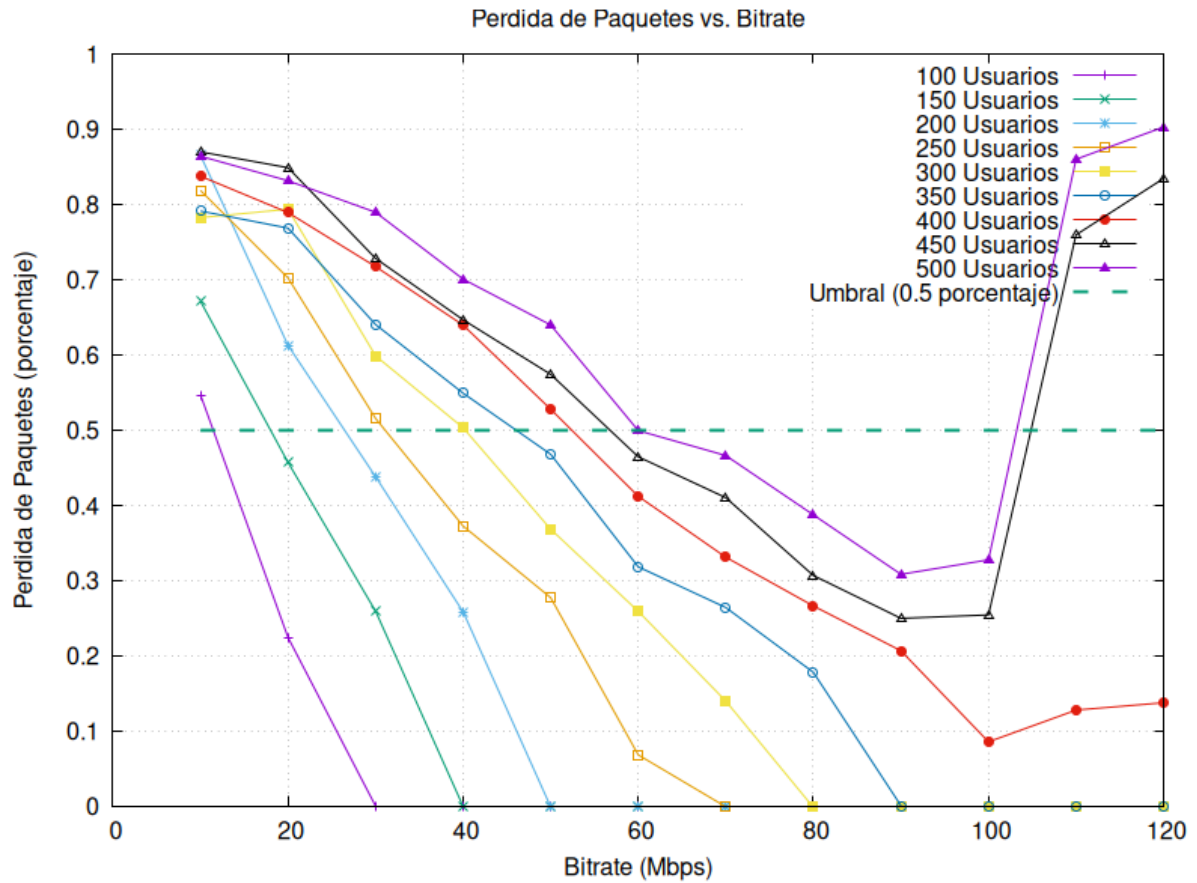
Gráficas de Retardo, jitter y pérdida de paquetes medios



(Figura 18) Gráfica retardo por cada conjunto de usuario//bitrate

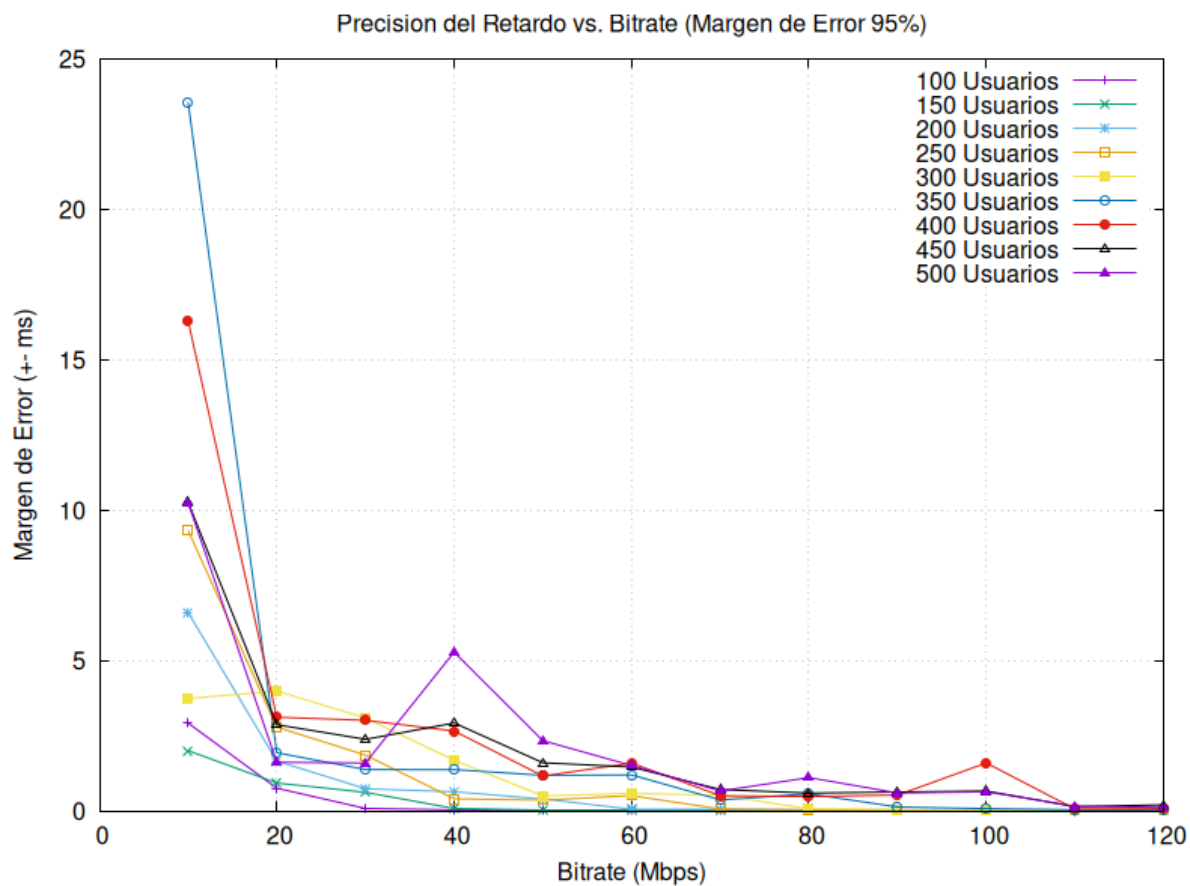


(Figura 19) Gráfica del jitter por cada conjunto de usuario//bitrate

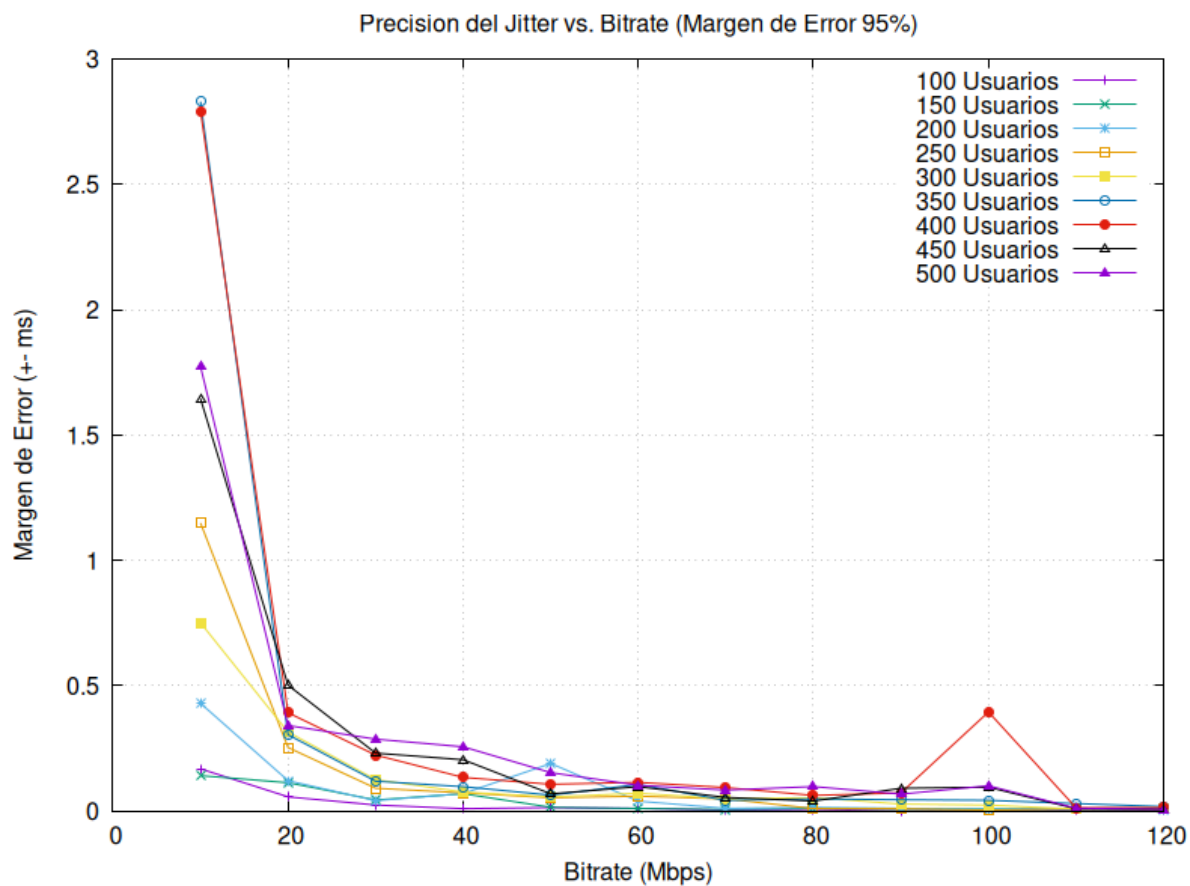


(Figura 20) Gráfica porcentaje de pérdida de paquetes por cada conjunto de usuarios//bitrate

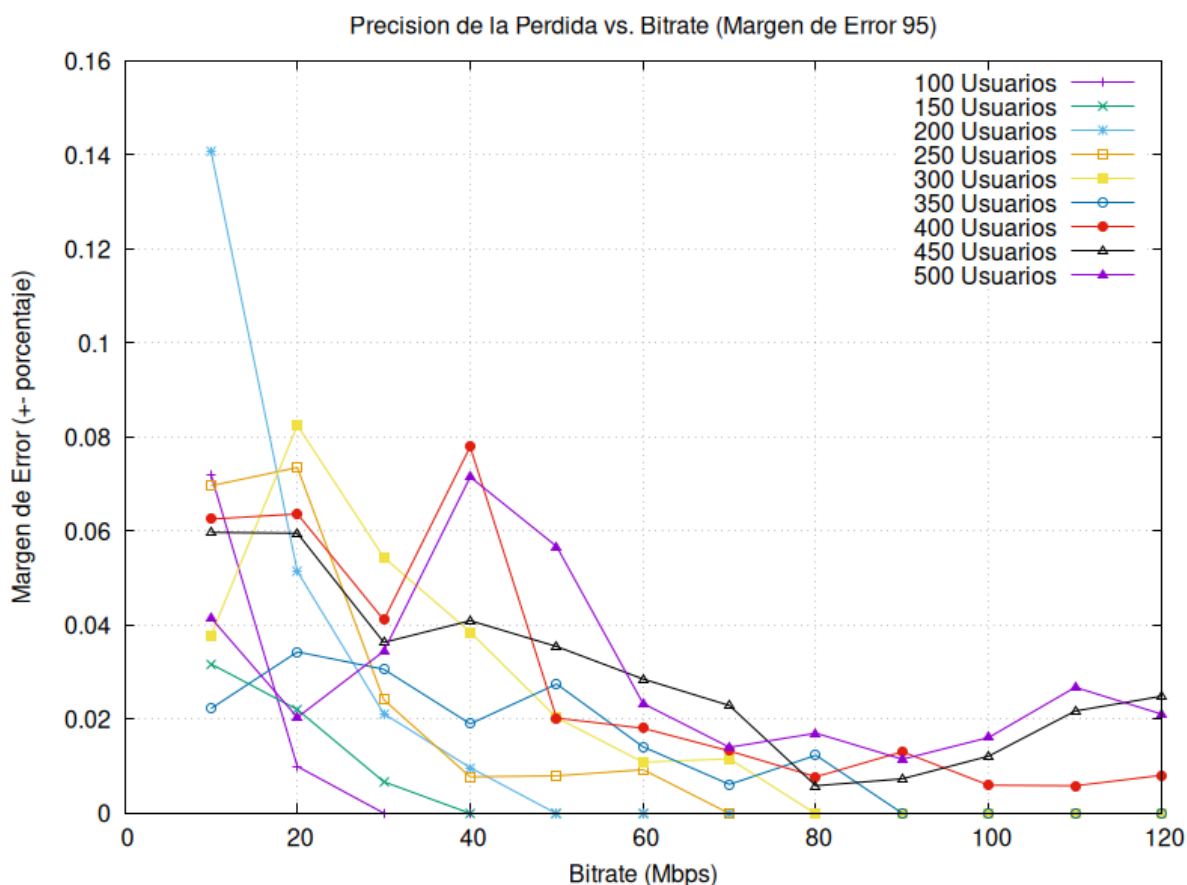
Gráficas de precisión de Retardo, jitter y pérdida de paquetes con margen de 95%



(Figura 21) Margen error de retardo



(Figura 22) Margen error de jitter



(Figura 23) Margen error pérdida de paquetes

Datos de la simulación en crudo: results.dat y sim-precision.dat

Los datos de simulación y estadísticos obtenidos se alojan en los ficheros results.dat y sim-precision.dat. El programa plot-results parte de ellos para generar todas las gráficas anteriores automáticamente.

Debido a su gran extensión hemos optado por dejar un enlace de descarga para los dos ficheros por si se quisieran consultar. Ambos ficheros se pueden abrir con cualquier editor de texto:

[ResultadosRaw-Simulación](#)

15. Analisis e interpretacion de los resultados

En este apartado se realiza un análisis profundo de los datos obtenidos en la fase de experimentación con ns-3. El objetivo es interpretar el comportamiento de la red para dar una respuesta directa a los objetivos del proyecto y, a su vez, discutir los

hallazgos clave sobre las limitaciones del sistema, como la identificación de cuellos de botella.

Objetivo del proyecto

Como se estableció en el apartado 4, el objetivo principal del proyecto es determinar el ancho de banda mínimo que StreamNow debe contratar para su enlace L1 (Valencia) con el fin de atender a una población de 100 a 500 usuarios. La condición fundamental es que dicho ancho de banda garantice el cumplimiento de los siguientes requisitos mínimos de Calidad de Servicio (QoS):

Parámetro	Valor máximo
Latencia (retardo)	20 ms
Jitter	8 ms
Pérdida de paquetes	0.5 %

Tabla 8 (Condiciones de calidad mínimas exigidas por la empresa)

Análisis de resultados obtenidos

Tras la fase de experimentación, en la que se realizaron diversas simulaciones y se obtuvieron los diferentes datos, se construyó la gráfica de la figura 15, donde se muestra que ancho de banda contratar a la operadora que da salida a Valencia según el número de usuarios en un rango de 100-500 usuarios, cumpliendo los requisitos de calidad antes mencionados.

Los datos que se obtuvieron se pueden resumir en la siguiente tabla:

Usuarios	Bitrate a contratar (Mbps)
100	30
150	40
200	50
250	70
300	80
350	90

400	100
450	No existe solución
500	No existe solución

Tabla 9 (Resultados característicos de la gráfica final)

Como se puede observar, la relación entre el número de usuarios y el bitrate requerido muestra una tendencia mayormente lineal en dos tramos definidos: de 100 a 200 usuarios y de 250 a 400. En estos segmentos, el ancho de banda necesario se incrementa en 10 Mbps por cada 50 usuarios adicionales.

Es destacable el salto que se produce entre los 200 y 250 usuarios, donde el bitrate requerido aumenta en 20 Mbps. Este punto de inflexión debe ser considerado en la planificación de capacidad de la operadora para evitar una posible saturación de la red.

Sin embargo, el hallazgo más significativo es la ausencia de una solución viable para 450 y 500 usuarios. Como se detalla a continuación, este resultado se debe a un cambio fundamental en el factor limitante de la red.

Limitaciones y cuellos de botella en la red

Al analizar las gráficas de rendimiento (figuras 14, 15 y 16), se puede explicar el comportamiento del sistema en sus diferentes fases:

- Rango de 100-400 usuarios: El enlace L1 como cuello de botella. En este rango, se observa una mejora directa en la latencia, el jitter y la pérdida de paquetes a medida que se incrementa el ancho de banda del enlace L1. Esto confirma que, para cargas bajas y medias de usuarios, el propio enlace L1 es el principal cuello de botella de la red. Un bitrate insuficiente en este enlace provoca la congestión que degrada las métricas, mientras que un aumento de su capacidad alivia directamente dicha congestión.
- A partir de 450 usuarios: El enlace LAN de Valencia como nuevo cuello de botella. Para cargas de 450 usuarios en adelante, el comportamiento del sistema cambia drásticamente. A pesar de seguir aumentando el bitrate del enlace L1, la pérdida de paquetes se dispara y se mantiene por encima del umbral del 0.5 %, imposibilitando encontrar una solución válida.

Esto se debe a que el cuello de botella principal se ha desplazado: ya no es el enlace L1, sino el enlace de 100 Mbps que conecta el Router 2 con la LAN

de Valencia. El router 2 comienza a recibir paquetes a un ritmo mayor del que su enlace de salida puede despachar. Esta acumulación sostenida provoca el desbordamiento de su búfer, resultando en un descarte masivo de paquetes. Este fenómeno, donde una capacidad de enlace inferior provoca la saturación de las colas, es una manifestación de bufferbloat.

Por lo tanto, se concluye que, a partir de 450 usuarios, aumentar la capacidad del enlace L1 deja de ser una solución efectiva, ya que el problema reside en otra parte de la infraestructura que ahora actúa como cuello de botella principal.

Soluciones para el bufferbloat

Existen dos soluciones destacables para solucionar el problema de bufferbloat. La primera consistiría en hablar con la operadora para que se aumenten las colas disponibles en los Routers que redirijan el tráfico a nuestra red (en concreto es relevante los que proceden de la región de Valencia ya que al haber más usuarios son los primeros en saturarse). La segunda solución sería aumentar la capacidad o el número de los servidores de caché, ya que esto reduciría el tráfico que va directamente a los servidores de StreamNow y que pasan por esos Routers críticos.

Dado que el problema de escalabilidad a partir de 450 usuarios es el bufferbloat en el Router 2, se proponen las siguientes soluciones para que StreamNow pueda seguir creciendo:

1. Aumentar la capacidad o números de los servidores de Caché: Aumentar la capacidad o el número de servidores de caché en la región de Valencia reduciría la cantidad de tráfico que necesita viajar hasta los servidores centrales. Esta es una solución altamente efectiva, ya que alivia la carga en toda la ruta de red, incluido el punto de congestión actual.
2. Implementar Gestión Activa de Colas (AQM): Se podría solicitar a la operadora la implementación de algoritmos de gestión activa de colas, como CoDel o FQ-CoDel, en los routers. Estos sistemas gestionan la longitud de las colas de forma inteligente para prevenir el bufferbloat y la pérdida masiva de paquetes, mejorando la calidad del servicio bajo altas cargas como en nuestro caso de estudio. Esto es principalmente relevante en la región de Valencia ya que es donde más usuarios (y por tanto más carga) se espera.

16. Conclusiones

Este proyecto ha cumplido con su objetivo principal: se ha determinado el ancho de banda que la empresa StreamNow necesita contratar para su enlace de Valencia en un rango de 100 a 500 usuarios, asegurando los requisitos mínimos de calidad establecidos. Los resultados obtenidos ofrecen una guía clara para la planificación de la red a corto y medio plazo.

Las conclusiones más importantes del estudio se pueden resumir en los siguientes puntos:

- **Guía de Ancho de Banda (hasta 400 usuarios):** Se ha comprobado que la red puede dar servicio de manera estable a una población de 100 a 400 usuarios. Para ello, el ancho de banda del enlace L1 debe escalar de forma progresiva, comenzando en 30 Mbps y llegando hasta los 100 Mbps.
- **Identificación del "Cuello de Botella Móvil":** El hallazgo más relevante es el descubrimiento de un "cuello de botella móvil".
 - Para cargas de hasta 400 usuarios, el rendimiento depende directamente del ancho de banda del enlace L1.
 - A partir de 450 usuarios, el problema se desplaza al enlace local de Valencia (con 100 Mbps de capacidad), que se satura y provoca un bufferbloat en el router que da acceso a la red de Valencia degradando abruptamente el servicio.

Como consecuencia del punto anterior, se concluye que con la infraestructura actual no es posible dar un servicio de calidad a 450 usuarios o más. Aumentar el ancho de banda del enlace L1 deja de ser una solución efectiva, ya que el problema reside en otro punto de la red.

Para que StreamNow pueda superar la barrera de los 450 usuarios a largo plazo se han recomendado implementar las siguientes soluciones:

- Mejorar el Sistema de Caché: Aumentar la capacidad o el número de servidores de caché. Esta medida es muy eficaz porque reduciría la cantidad de tráfico que viaja por la red principal, aliviando la carga sobre el punto de congestión. Relevante sobre todo en la red de Valencia.
- Implementar Gestión Activa de Colas (AQM): Solicitar a la operadora que implemente algoritmos de gestión de colas inteligentes (como CoDel o FQ-CoDel) previniendo el bufferbloat y la pérdida de paquetes, mejorando la estabilidad de la red cuando hay mucha carga.

17. Referencias

- [1] (PDF) Network Characteristics of Video Streaming Traffic
- [2] Predicting Session Length in Media Streaming
- [3] <https://www.idtexpress.com/es/blog/what-is-jitter-typical-causes-ways-to-reduce-it>
- [4] <https://obkio.com/blog/voip-latency/>
- [5] <https://www.pingplotter.com/wisdom/article/is-my-connection-good/>
- [6] <https://community.cisco.com/t5/webex-meetings-and-webex-app/acceptable-jitter-latency-and-packet-loss-for-audio-and-video-on-m-p/4301454>
- [7] https://en.wikipedia.org/wiki/Packet_loss
- [8] <https://www.wowza.com/blog/what-is-video-bitrate-and-what-bitrate-should-you-use>
- [9] https://en.wikipedia.org/wiki/Differentiated_services
- [10] https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_plcshp/configuration/xe-3s/qos-plcshp-xe-3s-book/qos-plcshp-pct-plc.html
- [11] <https://www.hughes.com/sites/hughes.com/files/2021-12/QoS-BB-Mgmt.pdf>
- [12] https://en.wikipedia.org/wiki/Random_early_detection
- [13] <https://dl.dell.com/topicspdf/poweredge-r740xd-reference-guide3-es-mx.pdf>
- [14] <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9500-series-switches/nb-06-cat9500-ser-data-sheet-cte-en.html>
- [15] <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9500-series-switches/nb-06-cat9500-ser-data-sheet-cte-en.html>
- [16] <https://ieeexplore.ieee.org/document/1040118/figures#figures>
- [17] <https://standards.ieee.org/ieee/802.3bj/5551/>
- [18] <https://datatracker.ietf.org/doc/html/rfc791>
- [19] <https://www.ietf.org/rfc/rfc793.txt>

- [20] <https://datatracker.ietf.org/doc/html/rfc2453>
- [21] <https://datatracker.ietf.org/doc/html/rfc2818>
- [22] <https://developers.google.com/media/vp9?hl=es-419>
- [23] <https://www.webmproject.org/>
- [24] <https://mpeg.chiariglione.org/standards/mpeg-4/audio.html>
- [25] https://books.google.es/books?id=1uxBwhAb_zYC
- [26] <https://www.cbtnuggets.com/blog/technology/networking/what-is-differentiated-services-code-point-dscp>
- [27] <https://www.cs.emory.edu/~cheung/Courses/558/Syllabus/11-Fairness/WFQ.html>
- [28] https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_regpkt/configuration/xr-16-8/qos-regpkt-xr-16-8-book/qos-regpkt-trfc.pdf
- [29] <https://docs.fortinet.com/document/fortigate/7.6.1/administration-guide/297431/traffic-shaping>
- [30] https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/qos/b-quality-of-service/m_qos-plcshp-trfc-plc-0.pdf
- [32] https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Video_codecs
- [33] <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>
- [34] <https://www.techtarget.com/>
- [35] <https://www.cloudflare.com/es-es/learning/>
- [36] <https://www.linuxjournal.com/article/2628>
- [37] <https://www.itprotoday.com/it-infrastructure/surfing-web-caching-technology-part-1>

© 2024, [Grupo 4 de Planificación y Simulación de Redes, de Ingeniería de Telecomunicación, US]. Todos los derechos reservados. Este documento no puede reproducirse, total ni parcialmente, sin la autorización previa de todos los integrantes del grupo.